

# iyzico API Document

---

## Authors:

Hakan ERDOĞAN

Sabri Onur TÜZÜN

**Version:** 1.3

## History of Changes

Date	Author	Version	Reason
28/09/15	Hakan ERDOGAN, Sabri Onur Tüzün	1.0	Draft document
02/05/16	Sabri Onur Tüzün	1.1	Pre Auth & Post Auth services support added
31/05/16	Emre Fındık	1.2	Node.js and Sandbox informations added

## Content

iyzico API Document.....	1
1 Overview.....	4
2 API Services.....	5
2.1 API Services Endpoints, Requests and Responses.....	7
3 Marketplace Solution .....	8
3.1 Buyer and Seller Contract .....	9
3.1.1 Alternative - 1: During registration.....	9
3.1.2 Alternative - 2: Product Listing or Check Out Page .....	10
3.2 SubMerchant Onboarding.....	10
3.2.1 New SubMerchant Onboarding.....	11
3.2.2 Update SubMerchant Info .....	18
3.3 Collection.....	21
3.3.1 Auth.....	21
3.3.2 Cancel.....	67
3.3.3 Refund.....	68
3.3.4 Pre Authorization.....	72
3.3.5 Post Authorization.....	81
3.4 Approval for Payout (payment transfer).....	87
3.4.1 Item Approval.....	87
3.4.2 Rollback item approval .....	89

## 1 Overview

iyzico API services are developed for all merchants that accept online payments. All services are developed based on stateless Restfull service architecture. API clients are required for communication such as data marshalling / unmarshalling and PKI (*public key infrastructure*) encryption / decryption.

iyzico provides API clients listed as below. Merchants should use appropriate clients based on their programming language.

- **Java**
- **.Net**
- **PHP**
- **Ruby on Rails (RoR)**
- **Phyton**
- **Node.js**

There are 3 main input fields for iyzico API integration.

- **API Base URL** : Main endpoint URL value for sandbox API client. For example `https://sandbox-api.iyzipay.com`
- **API Key** : Merchant unique identifier key. Merchants have to send this key for all sandbox requests. E.g: `sandbox-Txl2ctiktW6BrNmFo10pZiPn8W67Ksob`
- **Secret Key** : Merchant unique password for sandbox. Secret key is **never used** in data communication. E.g: `sandbox-GdzlHrXly6gQkq7NWOgx9DmXG`

It will be better to integrate iyzico API's initially in sandbox enviroment.

iyzico API services are developed for marketplace merchants and non-marketplace merchants. Non-marketplace merchants must consider API services endpoint specified below in blue.

## 2 API Services

iyzico API services endpoints consists of services shown in the table below. API clients that shared with merchants provide communication with API services endpoints such as data marshalling / unmarshalling and PKI (*public key infrastructure*) encryption / decryption.

Non-marketplace merchants must consider services listed below in blue. In additon to these, marketplace marchants must also consider API services shown below in red.

#	API Service (Endpoint)	API Service (Endpoint) URL	Method
1	Submerchant onboarding	/onboarding/submerchant	POST
2	Submerchant update	/onboarding/submerchant	PUT
3	Auth	/payment/iyzipos/auth/ecom	POST
4	Initialize 3D Secure	/payment/iyzipos/initialize3ds/ecom	POST
5	Auth 3D Secure	/payment/iyzipos/auth3ds/ecom	POST
6	Pre Auth	/payment/iyzipos/preauth/ecom	POST
7	Post Auth (Capture)	/payment/iyzipos/postauth	POST
8	Cancel	/payment/iyzipos/cancel	POST
9	Refund	/payment/iyzipos/refund	POST
10	Refund - Merchant commit	/payment/iyzipos/refund/merchant/charge	POST
11	Item approve	/payment/iyzipos/item/approve	POST
12	Item disapprove	/payment/iyzipos/item/disapprove	POST
13	Netting (crosbooking) – Money transfer to submerchant	/crossbooking/send	POST
14	Netting (crosbooking) – Money transfer from submerchant	/crossbooking/receive	POST
15	Healthcheck	/payment/test	GET
16	BIN Check	/payment/bin/check	POST
17	Installement	/payment/iyzipos/installment	POST
18	Card Storage	/cardstorage/card	POST
19	Delete stored card	/cardstorage/card	DELETE
20	Get stored card info	/cardstorage/cards	POST
21	Payout completed transactions	/reporting/settlement/payoutcompleted	POST
22	Payout bounced	/reporting/settlement/bounced	POST

## 2.1 API Services Endpoints, Requests and Responses

iyzico clients are required to access iyzico API services endpoints. iyzico clients provides authorization (PKI) for merchants and make communication easier between API services endpoints.

In all API services endpoints requests,

- Authorization and x-iyzi-rnd values must send within Request Header. API clients perform it on your behalf.
- Values listed below send as optional.

Input Name	Data Type	Mandatory	Explanation
<a href="#">locale</a>	String	No	For language settings.Default value is <b>tr</b>
<a href="#">conversationId</a>	String	No	Optional parameter to match request and response

- In the below table, you will see common parameters in all responses. If the result is successful (*status=success*) then other fields will be filled based on the request.

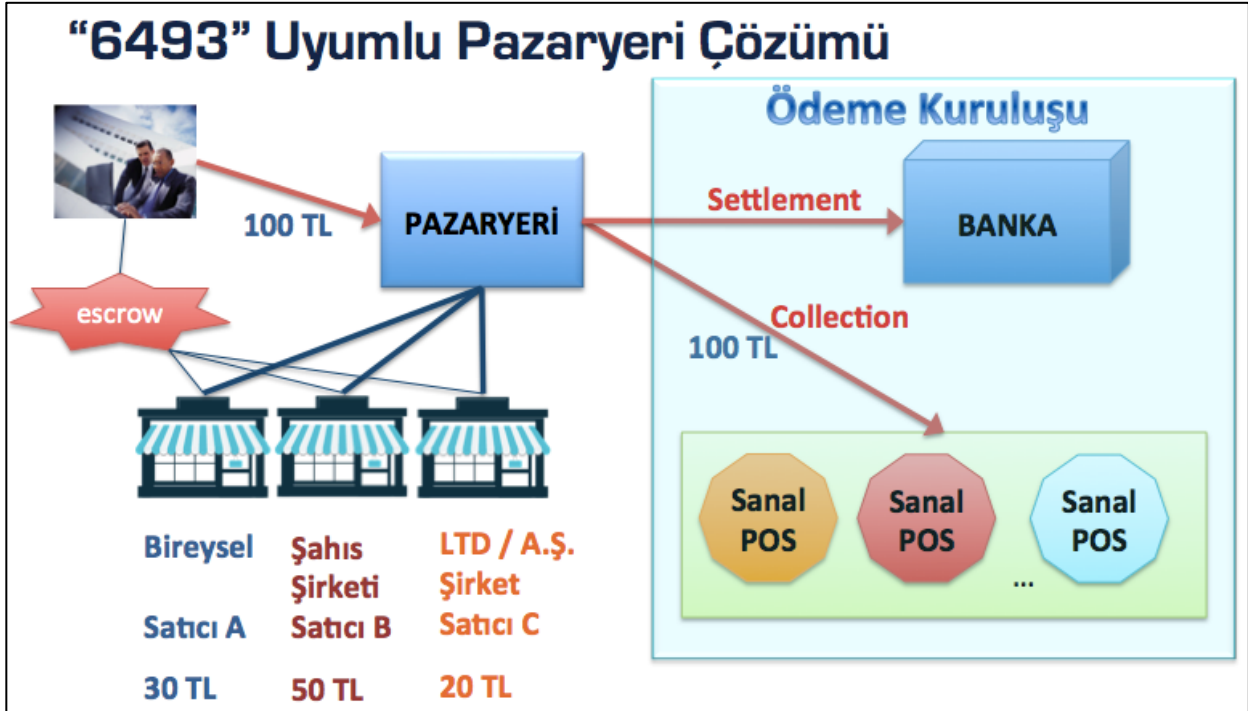
Output Name	Data Type	Explanation
<a href="#">status</a>	String	Service response result. success / failure
<a href="#">errorCode</a>	String	Error code if service response status is failure
<a href="#">errorMessage</a>	String	Error message if service response status is failure.
<a href="#">errorGroup</a>	String	Error group if service response status is failure.
<a href="#">locale</a>	String	For language settings.Default value is <b>tr</b>
<a href="#">conversationId</a>	String	If sent, optional parameter to match request and response
<a href="#">systemTime</a>	Long	Response system timestamp value

### 3 Marketplace Solution

Please check the example shown below, to better understand the marketplace solution,

Example :

A,B,C are submerchants of a marketplace. A is a person, B is private company and C is a company. Buyer's basket contains 3 different items. First item's price is 30 TL and belongs to seller A, second item's price is 50 TL and belongs to seller B and last item's price is 20 TL and belongs to seller C.



Escrow will perform for each product. Seller or status of each product may be different.

E.g. seller A sent product, buyer received it and approve the item. Seller B sent product to buyer but buyer didn't accept item and return it. Seller C didn't send the product yet.

Merchants must trigger escrow by using itemApproval API Service. iyzico will manage settlement between merchants and submerchants when merchant have complete escrow, till that all payments will keep in secure bank account by iyzico.



Marketplace solution contains collection and settlement processes. There are 3 main steps for integration flow :

- Submerchant onboarding
- Payment
- Item approval

Merchants have to register their submerchants by using [/onboarding/submerchant](#) API service endpoints. Merchants receive subMerchantKey parameter within response for each subMerchant onboarding request. Merchants have to send subMerchantKey and subMerchantPrice parameters within payment request.

Before starting integration, buyers and sellers must approve iyzico contract.

### 3.1 Buyer and Seller Contract

iyzico manage payment system between sellers and buyers, that is why both of them must approve the contracts that shown in the links. These actions must be taken for one time each buyer and seller.

- For buyers, <https://www.iyzico.com/pazaryeri-alici-anlasma/>
- For sellers <https://www.iyzico.com/pazaryeri-satici-anlasma/>

Approval of iyzico terms and conditions can be taken in ,

- Registration page
- Product listing page or check out page

#### 3.1.1 Alternative - 1: During registration

“iyzico platform user agreement” is able to displayed on registration page and make submerchants agree this agreement to get involved to marketplace solution. It is possible to access agreement from <https://www.iyzico.com/pazaryeri-anlasma>

### 3.1.2 Alternative - 2: Product Listing or Check Out Page

#### 3.1.2.1 *iyzico Account for Seller:*



It is possible to make submerchant approve the agreement during product or service listing process. <https://www.iyzico.com/pazaryeri-anlasma>

#### 3.1.2.2 *iyzico Account for Buyer:*

For buyers it is enough to maket he approve the agreement once.

Kart Üzerindeki Ad Soyad

Kart Numarası

05 2015 CVV

[iyzico Platform Kullanım Sözleşmesini okudum ve kabul ediyorum.](#) ?

TOPLAM: 10,00 TL

Ödeme Yap

## 3.2 SubMerchant Onboarding

Marketplace merchants must register their submerchants with accurate informations such as IBAN which is required for settlement process, by using subMerchant onboarding API service endpoint. After registration if there is any update on subMerchant information, merchants should update subMerchant account by using subMerchant onboarding API service endpoint.

### 3.2.1 New SubMerchant Onboarding

Base on legal structure there is 3 subMerchant type,

- person,
- private company,
- company.

Each of them have different parameters during onboarding. iyzico onboarding API service endpoint returns a unique subMerchantKey parameter within response for each subMerchant.

subMerchantKey is a mandatory parameter within payment request for marketplace solution. That is why merchants must keep all subMerchantKey values for each subMerchant in their system.

**Note:** All parameters send by merchants should be valid and accurate.

#### 3.2.1.1 Person – SubMerchant Onboarding

If subMerchant is a person, required parameters listed below for registration.

- API Service (Endpoint) Name: **/onboarding/submerchant**
- Method: **HTTP POST**

Input Name	Data Type	Mandatory	Explanation
name	String	No	SubMerchant Name
gsmNumber	String	No	SubMerchant telephone number
contactName	String	Yes	
contactSurname	String	Yes	
email	String	Yes	Submerchant email address
address	String	Yes	Submerchant address
iban	String	No **	IBAN – bank account number of subMerchant. It will send to bank with contactName and contactSurname values. So owner of the IBAN have to match with contactName and contactSurname
subMerchantExternalId	String	Yes	Submerchant ID in Merchant's system
identityNumber	String	No **	Citizen number
subMerchantType	String	Yes	<b>PERSONAL</b> enum value

Response value :

Output Name	Data Type	Explanation
subMerchantKey	String	Unique value for each subMerchant

### Request Example (Java) :

```

ClientConfiguration configuration = new ClientConfiguration();
configuration.setApiKey("api key");
configuration.setSecretKey("secret key");
configuration.setBaseUrl("https://sandbox-api.iyzipay.com");

OnboardingServiceClient client = new OnboardingServiceClientImpl(configuration);

CreateSubMerchantRequest request = new CreateSubMerchantRequest();
request.setLocale(RequestLocaleType.TR.getValue());
request.setConversationId("123456789");
request.setSubMerchantExternalId("B49220");
request.setSubMerchantType(SubMerchantRequestType.PERSONAL.name());
request.setAddress("Nidakule Göztepe İş Merkezi Merdivenköy Mah. Bora Sok. No:1 Kat:19 Bağımsız
70/73 Göztepe Kadıköy 34732");
request.setContactName("Sabri Onur");
request.setContactSurname("Tüzün");
request.setEmail("email@submerchantemail.com");
request.setGsmNumber("+905350000000");
request.setName("Sabri Onur'un mağazası");
request.setIban("TR180006200119000006672315");
request.setIdentityNumber("31300864726");

CreateSubMerchantResponse response = client.createSubMerchant(request);

System.out.println("conversationId="+response.getConversationId());
System.out.println("locale="+response.getLocale());
System.out.println("status="+response.getStatus());
System.out.println("errorCode="+response.getErrorCode());
System.out.println("errorMessage="+response.getErrorMessage());
System.out.println("errorGroup="+response.getErrorGroup());
System.out.println("systemTime="+response.getSystemTime());
System.out.println("subMerchantKey="+response.getSubMerchantKey());

```

### Response Example (Java)

```

conversationId=123456789
locale=tr
status=success
errorCode=null

```

```

errorMessage=null
errorGroup=null
systemTime=1443687828497
subMerchantKey=|4FCFyclof0paTP6687dOoch9Po=

```

#### JSON format of request example:

```

{
  "locale" : "tr",
  "conversationId" : "123456789",
  "name" : "Sabri Onur'un mağazası",
  "email" : "email@submerchantemail.com",
  "gsmNumber" : "+905350000000",
  "address" : "Nidakule Göztepe İş Merkezi Merdivenköy Mah. Bora Sok. No:1 Kat:19 Bağımsız 70/73
  Göztepe Kadıköy 34732",
  "iban" : "TR180006200119000006672315",
  "contactName" : "Sabri Onur",
  "contactSurname" : "Tüzün",
  "subMerchantExternalId" : "B49221",
  "identityNumber" : "31300864726",
  "subMerchantType" : "PERSONAL"
}

```

#### JSON format of response example:

```

{
  "status" : "success",
  "locale" : "tr",
  "systemTime" : 1443687828497,
  "conversationId" : "123456789",
  "subMerchantKey" : "|4FCFyclof0paTP6687dOoch9Po="
}

```

#### 3.2.1.2 Private Company - SubMerchant Onboarding

If subMerchant is a private company, required parameters listed below for registration.

- API Service (Endpoint) Name: **/onboarding/submerchant**
- Method: **HTTP POST**

Input name	Data Type	Mandatory	Explanation
name	String	Hayır	SubMerchant Name
gsmNumber	String	Hayır	SubMerchant telephone number
taxOffice	String	Evet	Submerchant Tax Office
legalCompanyTitle	String	Evet	Legal entity name of subMerchant
email	String	Evet	Submerchant email
address	String	Evet	Submerchant address
iban	String	Hayır **	IBAN – bank account number of subMerchant. It will send to bank with legalCompanyTitle values.
subMerchantExternalId	String	Evet	Submerchant ID in Merchant's system
identityNumber	String	Hayır **	Citizen Number
subMerchantType	String	Evet	<b>PRIVATE_COMPANY</b> enum value

Response value :

Output Name	Data Type	Explanation
subMerchantKey	String	Unique value for each subMerchant

#### Request Example (Java):

```
ClientConfiguration configuration = new ClientConfiguration();
configuration.setApiKey("api key");
configuration.setSecretKey("secret key");
configuration.setBaseUrl("https://sandbox-api.iyzipay.com");

OnboardingServiceClient client = new OnboardingServiceClientImpl(configuration);

CreateSubMerchantRequest request = new CreateSubMerchantRequest();
request.setLocale(RequestLocaleType.TR.getValue());
request.setConversationId("123456789");
request.setSubMerchantExternalId("S49221");
request.setSubMerchantType(SubMerchantRequestType.PRIVATE_COMPANY.name());
request.setAddress("Nidakule Göztepe İş Merkezi Merdivenköy Mah. Bora Sok. No:1 Kat:19 Bağımsız 70/73 Göztepe Kadıköy 34732");
request.setTaxOffice("Kadıköy V.D.");
request.setLegalCompanyTitle("Sabri Onur Tüzün Bilişim Hizmetleri");
request.setEmail("email@submerchantemail.com");
request.setGsmNumber("+905350000000");
request.setName("Sabri Onur'un mağazası");
request.setIban("TR180006200119000006672315");
request.setIdentityNumber("31300864726");

CreateSubMerchantResponse response = client.createSubMerchant(request);
```

```
System.out.println("conversationId="+response.getConversationId());
System.out.println("locale="+response.getLocale());
System.out.println("status="+response.getStatus());
System.out.println("errorCode="+response.getErrorCode());
System.out.println("errorMessage="+response.getErrorMessage());
System.out.println("errorGroup="+response.getErrorGroup());
System.out.println("systemTime="+response.getSystemTime());
System.out.println("subMerchantKey="+response.getSubMerchantKey());
```

#### Response Example (Java):

```
conversationId=123456789
locale=tr
status=success
errorCode=null
errorMessage=null
errorGroup=null
systemTime=1443788792146
subMerchantKey=lcB6bA9DWEF5e8yGJI0uIEcW1kw=
```

#### JSON format of request example:

```
{
  "locale" : "tr",
  "conversationId" : "123456789",
  "name" : "Sabri Onur'un mağazası",
  "email" : "email@submerchantemail.com",
  "gsmNumber" : "+905350000000",
  "address" : "Nidakule Göztepe İş Merkezi Merdivenköy Mah. Bora Sok. No:1 Kat:19 Bağımsız 70/73
  Göztepe Kadıköy 34732",
  "iban" : "TR180006200119000006672315",
  "taxOffice" : "Kadıköy V.D.",
  "legalCompanyTitle" : "Sabri Onur Tüzün Bilişim Hizmetleri",
  "subMerchantExternalId" : "S49221",
  "identityNumber" : "31300864726",
  "subMerchantType" : "PRIVATE_COMPANY"
}
```

### JSON format of response example:

```
{
  "status": "success",
  "locale": "tr",
  "systemTime": 1443788792146,
  "conversationId": "123456789",
  "subMerchantKey": "IcB6bA9DWEF5e8yGJI0uIEcW1kw="
}
```

#### 3.2.1.3 Company – SubMerchant Onboarding

If subMerchant is a company, required parameters listed below for registration.

- API Service (Endpoint) Name: **/onboarding/submerchant**
- Method: **HTTP POST**

Input Name	Data Type	Mandatory	Explanation
name	String	Hayır	SubMerchant Name
gsmNumber	String	Hayır	SubMerchant telephone number
taxOffice	String	Evet	SubMerchant tax Office
taxNumber	String	Evet	Alt SubMerchant tax number
legalCompanyTitle	String	Evet	Legal entity of subMerchant
email	String	Evet	Submerchant email
address	String	Evet	Submerchant address
iban	String	Hayır **	A IBAN – bank account number of subMerchant. It will send to bank with legalCompanyTitle value.
subMerchantExternalId	String	Evet	Submerchant ID in Merchant's system
subMerchantType	String	Evet	<b>LIMITED_OR_JOINT_STOCK_COMPANY</b> enum value

Response Value :

Output Name	Data Type	Explanation
subMerchantKey	String	Unique value for each subMerchant

### Request Example (Java) :

```
ClientConfiguration configuration = new ClientConfiguration();
configuration.setApiKey("api key");
configuration.setSecretKey("secret key");
configuration.setBaseUrl("https://sandbox-api.iyzipay.com");
```

```
OnboardingServiceClient client = new OnboardingServiceClientImpl(configuration);
```



```

CreateSubMerchantRequest request = new CreateSubMerchantRequest();
request.setLocale(RequestLocaleType.TR.getValue());
request.setConversationId("123456789");
request.setSubMerchantExternalId("AS49222");
request.setSubMerchantType(SubMerchantRequestType.LIMITED_OR_JOINT_STOCK_COMPANY.name());
request.setAddress("Nidakule Göztepe İş Merkezi Merdivenköy Mah. Bora Sok. No:1 Kat:19 Bağımsız 70/73 Göztepe Kadıköy 34732");
request.setTaxOffice("Kadıköy V.D.");
request.setTaxNumber("9261877");
request.setLegalCompanyTitle("XYZ Bilişim Hizmetleri A.Ş.");
request.setEmail("email@submerchantemail.com");
request.setGsmNumber("+905350000000");
request.setName("Sabri Onur'un mağazası");
request.setIban("TR180006200119000006672315");

CreateSubMerchantResponse response = client.createSubMerchant(request);

System.out.println("conversationId=" + response.getConversationId());
System.out.println("locale=" + response.getLocale());
System.out.println("status=" + response.getStatus());
System.out.println("errorCode=" + response.getErrorCode());
System.out.println("errorMessage=" + response.getErrorMessage());
System.out.println("errorGroup=" + response.getErrorGroup());
System.out.println("systemTime=" + response.getSystemTime());
System.out.println("subMerchantKey=" + response.getSubMerchantKey());

```

### Response Example (Java):

```

conversationId=123456789
locale=tr
status=success
errorCode=null
errorMessage=null
errorGroup=null
systemTime=1443790256853
subMerchantKey=D9V/MqIRitUzA4dutL+nCBvnWfs=

```

#### JSON format of request example:

```
{
  "locale" : "tr",
  "conversationId" : "123456789",
  "name" : "Sabri Onur'un mağazası",
  "email" : "email@submerchantemail.com",
  "gsmNumber" : "+905350000000",
  "address" : "Nidakule Göztepe İş Merkezi Merdivenköy Mah. Bora Sok. No:1 Kat:19 Bağımsız 70/73
  Göztepe Kadıköy 34732",
  "iban" : "TR180006200119000006672315",
  "taxOffice" : "Kadıköy V.D.",
  "legalCompanyTitle" : "XYZ Bilişim Hizmetleri A.Ş.",
  "subMerchantExternalId" : "AS49222",
  "taxNumber" : "9261877",
  "subMerchantType" : "LIMITED_OR_JOINT_STOCK_COMPANY"
}
```

#### JSON format of response example:

```
{
  "status" : "success",
  "locale" : "tr",
  "systemTime" : 1443790256853,
  "conversationId" : "123456789",
  "subMerchantKey" : "D9V/MqIRitUzA4dutL+nCBvnWfs="
}
```

### 3.2.2 Update SubMerchant Info

Merchants can update SubMerchant's informations by using subMerchant onboarding API service endpoint.

**Note 1:** subMerchantKey is a mandatory value in update request

**Note 2:** Valid IBAN is a mandatory value in update request.

**Note 3:** subMerchant type is not updatable. If type of any subMerchant is changed, merchants must register it as another subMerchant. E.g. Seller A is a person and switch company type to private company. Merchant must register Seller A again with new company type parameter.

- API Service (Endpoint) Name: **/onboarding/submerchant**
- Method: **HTTP PUT**

Input Name	Data Type	Mandatory	Explanation
<b>subMerchantKey</b>	String	<b>Evet</b>	
<b>iban</b>	String	<b>Evet</b>	IBAN – bank account number of subMerchant. It will send to bank with contactName and contactSurname value.
<b>name</b>	String	Hayır	SubMerchant Name
<b>gsmNumber</b>	String	Hayır	SubMerchant telephone number
<b>contactName</b>	String	Evet	Submerchant Name
<b>contactSurname</b>	String	Evet	Submerchant surname
<b>email</b>	String	Evet	Submerchant email
<b>address</b>	String	Evet	Submerchant address
<b>identityNumber</b>	String	Hayır **	Citizen number
<b>subMerchantType</b>	String	Evet	<b>PERSONAL</b> enum value

#### Request Example (Java):

```
ClientConfiguration configuration = new ClientConfiguration();
configuration.setApiKey("api key");
configuration.setSecretKey("secret key");
configuration.setBaseUrl("https://sandbox-api.iyzipay.com");

OnboardingServiceClient client = new OnboardingServiceClientImpl(configuration);

UpdateSubMerchantRequest request = new UpdateSubMerchantRequest();
request.setLocale(RequestLocaleType.TR.getValue());
request.setConversationId("123456789");
request.setSubMerchantKey("l4FCFyclof0paTP6687dOoch9Po=");
request.setIban("TR630006200027700006678204");
request.setAddress("Nidakule Göztepe İş Merkezi Merdivenköy Mah. Bora Sok. No:1 Kat:19 Bağımsız 70/73 Göztepe Kadıköy 34732");
request.setContactName("Hakan");
request.setContactSurname("Erdoğan");
request.setEmail("email@submerchantemail.com");
request.setGsmNumber("+905350000000");
request.setName("Hakan Erdoğan'in mağazası");
request.setIdentityNumber("31300864726");

UpdateSubMerchantResponse response = client.updateSubMerchant(request);

System.out.println("conversationId=" + response.getConversationId());
System.out.println("locale=" + response.getLocale());
```

```
System.out.println("status=" + response.getStatus());
System.out.println("errorCode=" + response.getErrorCode());
System.out.println("errorMessage=" + response.getErrorMessage());
System.out.println("errorGroup=" + response.getErrorGroup());
System.out.println("systemTime=" + response.getSystemTime());
```

#### Response Example (Java):

```
conversationId=123456789
locale=tr
status=success
errorCode=null
errorMessage=null
errorGroup=null
systemTime=1444129731762
```

#### JSON format of request example:

```
{
  "locale" : "tr",
  "conversationId" : "123456789",
  "name" : "Hakan Erdoğan'ın mağazası",
  "email" : "email@submerchantemail.com",
  "gsmNumber" : "+905350000000",
  "address" : "Nidakule Göztepe İş Merkezi Merdivenköy Mah. Bora Sok. No:1 Kat:19 Bağımsız 70/73
Göztepe Kadıköy 34732",
  "iban" : "TR630006200027700006678204",
  "contactName" : "Hakan",
  "contactSurname" : "Erdoğan",
  "subMerchantKey" : "l4FCFyclof0paTP6687dOoch9Po=",
  "identityNumber" : "31300864726"
}
```

#### JSON format of response example:

```
{
  "status" : "success",
  "locale" : "tr",
  "systemTime" : 1444129731762,
```

```
"conversationId" : "123456789"
```

```
}
```

### 3.3 Collection

Auth (3DS – non 3DS), preauth, refund and cancel are common API Services for marketplace and non-marketplace merchants.

#### 3.3.1 Auth

It can be 3DS or non-3DS depends on merchant's choice. Non-3DS payments has resulted in a simple request/response cycle.

##### 3.3.1.1 *Payment Group, Product Type, Main Payment Value*

Payment group types are listed below. There is not difference in payment process between listing and subscription. But payment process of product is different from them.

- PRODUCT
- LISTING
- SUBSCRIPTION

productType type are listed below:

- VIRTUAL
- PHYSICAL

ShippingAddress is an optional parameter for VIRTUAL products. If productType of any item's in the basket is PHYSICAL, shippingAddress will be mandatory parameter in payment request.

#### **For Marketplace Merchants;**

- Product type will be VIRTUAL for LISTING and SUBSCRIPTION payment groups.
- Product type may be VIRTUAL or PHYSICAL for PRODUCT payment groups.
- subMerchantPrice and subMerchantKey parameters must send within payment request if payment group is PRODUCT.
- Escrow process will performed, if payment group is PRODUCT. (Till item approval accomplish payments keep in secure bank account of iyzico.)

- Settlement will performed without any item approval process if payment group is LISTING or SUBSCRIPTION.
- Settlement will perform regarding to payout period after item approval process is accomplished.

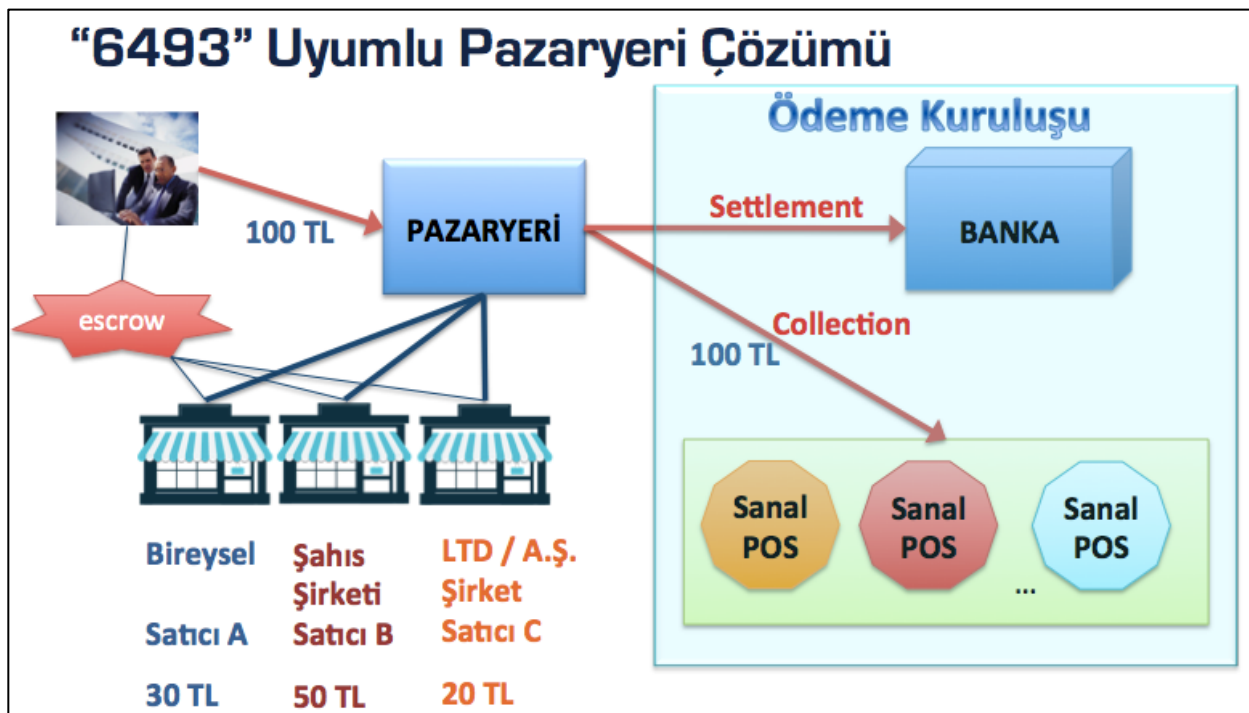
### Non-Marketplace Merchants;

- Product type must be VIRTUAL for LISTING and SUBSCRIPTION payment groups.
- Product type may be VIRTUAL or PHYSICAL for PRODUCT payment groups.
- There is no need to send subMerchantPrice and subMerchantKey parameters within payment request if payment group is PRODUCT.
- There is no Escrow process.
- Settlement will perform regarding to payout period, without any item approval process.

### Example – Payment Process for PRODUCT (Marketplace Merchants):

Basket contains 3 items from 3 different subMerchants

- Item of SubMerchant A is 30 TL
- Item of SubMerchant B is 50 TL
- Item of SubMerchant C is 20 TL



Merchants send card details, address details, payment details within payment request. In addition to these parameters,

- Basket volume (**price**).
- Final price that will be captured from card. (**paidPrice**).
- For each item in basket ;
  - Item price (**price**),
  - Payout amount to subMerchant (**subMerchantPrice**),
  - SubMerchant ID (**subMerchantKey**).

Regarding to example above,

- ...
- price = 100
- paidPrice=110
  - basketItem 1
    - price=30
    - subMerchantPrice=27
    - subMerchantKey=G2FCFyclof0paTP6687dOoch9Tc=
  - basketItem 2
    - price=50
    - subMerchantPrice=42
  - subMerchantKey=D9V/MqIRitUzA4dutL+nCBvnWfs=
  - basketItem 3
    - price=20
    - subMerchantPrice=18
    - subMerchantKey=D9V/MqIRitUzA4dutL+nCBvnWfs=
- ...

Response parameters,

- **paymentId** for successful payment request
- **paymentTransactionId** for each item in basket.

Merchants have to keep both of them (paymentId and paymentTransactionId) in their system .

- **paymentId**: Mandatory value to cancel payment.
- **transactionId**: Mandatory value to item approval, and refund transaction.

### 3.3.1.2 Payment Request and Response Datas

Input İsmi	Data Type	Mandatory	Explanation
price	Decimal	Yes	Total amount of basket. The sum of basket items' amount must be equal to the price of basket
paidPrice	Decimal	Yes	Final amount that will be captured from customer's card. including installment fee.
currency	String	Yes	Default is TRY. You can set USD, EUR, GBP and IRR as value.
installment	Integer	Yes	Installment value. For single installment payments, it should be 1. (Valid values: 1, 2, 3, 6, 9)
basketId	String	No	Merchant's basket ID.
paymentChannel	String	No	Payment channel. WEB, MOBILE, MOBILE_WEB, MOBILE_IOS, MOBILE_ANDROID, MOBILE_WINDOWS, MOBILE_TABLET, MOBILE_PHONE
paymentGroup	String	No	Default value is PRODUCT. Valid values are PRODUCT, LISTING, SUBSCRIPTION
paymentSource	String	Yes	Valid values are: SHOPIFY, MAGENTO, PRESTASHOP, WOOCOMMERCE, OPENCART
cardNumber	String	Yes **	15 (amex) or 16 (visa, mc) digits card number
expireYear	String	Yes **	Expire year of the card
expireMonth	String	Yes **	Expire month of the card as 2 digits
cvc	String	Yes **	4 (amex) or 3 (visa, mc) digits card verification code
cardHolderName	String	Yes **	Name of the card owner
callbackUrl	String	No **	Merchant callback url for payment result notification
registerCard	Integer	No	Parameter that shows card details will store or not. Default value is 0 and valid values are 0 or 1.
cardAlias	String	No	If registerCard value is 1 cardAlias can send within request.
cardUserKey	String	No **	Special for card user, that will return to registerCard request. It is mandatory value if payment will process with stored card.
cardToken	String	No **	Special for card, that will return to registerCard request. It is mandatory value if payment will process with stored card.
id (Buyer)	String	Yes	Buyer ID
name (Buyer)	String	Yes	Buyer name
surname (Buyer)	String	Yes	Buyer surname
identityNumber (Buyer)	String	Yes	Buyer identity number
city (Buyer)	String	Yes	Buyer city value
country (Buyer)	String	Yes	Buyer country value
email (Buyer)	String	Yes	Buyer email address
gsmNumber (Buyer)	String	No	Buyer telephone number



ip (Buyer)	String	Yes	Buyer IP address
registrationAddress (Buyer)	String	Yes	Buyer registration address
zipCode (Buyer)	String	No	Buyer zip code
registrationDate (Buyer)	String	No	Buyer registration date. Date format should be 2015-09-17 23:45:06
lastLoginDate (Buyer)	String	No	Date format should 2015-09-17 23:45:06
contactName (BillingAddress)	String	Yes	Contact name of billing address.
city (BillingAddress)	String	Yes	City name of billing address.
country (BillingAddress)	String	Yes	Country name of billing address.
address (BillingAddress)	String	Yes	Address details of billing address.
zipCode (BillingAddress)	String	No	Zip code of billing address
contactName (ShippingAddress)	String	Yes **	Contact name of shipping address. If one of the itemType in basket is PHYSICAL, then this parameter will mandatory
city (ShippingAddress)	String	Yes **	City name of shipping address. If one of the itemType in basket is PHYSICAL, then this parameter will mandatory
country (ShippingAddress)	String	Yes **	Country name of shipping address. If one of the itemType in basket is PHYSICAL, then this parameter will mandatory
address (ShippingAddress)	String	Yes **	Address details of shipping address. If one of the itemType in basket is PHYSICAL, then this parameter will mandatory
zipCode (ShippingAddress)	String	No	Zip code of shipping address.
id (BasketItem)	String	Yes	Basket Item ID
itemType (BasketItem)	String	Yes	Valid values are PHYSICAL,VIRTUAL
name (BasketItem)	String	Yes	Name of the item in the basket
category1 (BasketItem)	String	Yes	Category 1 for item in the basket.
category2 (BasketItem)	String	No	Category 2 for item in the basket.
price (BasketItem)	Decimal	Yes	Price value for each item. Must be greater than zero.
subMerchantPrice (BasketItem)	Decimal	Yes **	Payout amount to subMerchant of each item. Must be greater than zero. Mandatory value for marketplace solution
subMerchantKey (BasketItem)	String	Yes **	Submerchant ID value. Mandatory value for marketplace solution

Output İsmi	Veri Tipi	Açıklama
paymentId	String	Id of the payment. Merchants should keep payment ID in their system. ( <i>This id will be used for cancel requests</i> )
price	Decimal	Total amount of basket. The sum of basket items' amount must be equal to the price of basket
paidPrice	Decimal	Final price that will be captured from card. Merchants should store this value in their system.
paymentStatus	String	Shows the status of payment transaction. Valid values are;SUCCESS, FAILURE, INIT_THREEDS, CALLBACK_THREEDS, BKM_POS_SELECTED, CALLBACK_PECCO
installment	Integer	Installment value. For single installment payments, it should be 1. (Valid values: 1, 2, 3, 6, 9)
basketId	String	Merchant's basket ID.
binNumber	String	First 6 digit of card
cardAssociation	String	Valid values are VISA, MASTER_CARD, AMERICAN_EXPRESS
cardFamily	String	Valid values are Bonus, Axess, World, Maximum, Paraf, CardFinans, AsyaCard, ...
cardType	String	Valid values are CREDIT_CARD, DEBIT_CARD, PREPAID_CARD
cardUserKey	String	In case of card detail stored, cardUserKey will return in response value. It is special for user. Mandatory value in case of a payment that will perform with stored card.
cardToken	String	In case of card detail stored, cardUserKey will return in response value. It is special for Card. Mandatory value in case of a payment that will perform with stored card.
fraudStatus	Integer	Fraud risk status for payment request. Valid values are -1, 0, 1. <i>This will always return 1 for checkout form solution</i>
iyziCommissionFee	Decimal	iyzico commission fee for payment transaction
iyziCommissionRateAmount	Decimal	iyzico commission rate amount for payment transaction
merchantCommissionRate	Decimal	E.g. example price=100, paidPrice=110 merchant commission rate is %10

merchantCommissionRateAmount	Decimal	E.g. price=100, paidPrice=110 merchant commission rate amount is 10
paymentTransactionId (ItemTransactions)	String	Ödeme kırılımına ait id, üye işyeri tarafından mutlaka saklanmalıdır. Ödeme kırılımının iadesi, onayı, onay geri çekmesi ve iyzico ile iletişimde kullanılır. Tercihen itemId ile ilişkili bir şekilde tutulmalıdır
itemId (ItemTransactions)	String	item ID of each item in basket
price (ItemTransactions)	Decimal	item price of each item in basket
paidPrice (ItemTransactions)	Decimal	Captured amount from card based on each items. Merchants should keep paidPrice in their system.
transactionStatus (ItemTransactions)	Integer	Status of a payment transaction. Valid values are 0, -1, 1, 2. <i>This will always return 2 for checkout form solution</i>
blockageRate (ItemTransactions)	Decimal	Blockage rate based on each item.
blockageRateAmountMerchant (ItemTransactions)	Decimal	Blockage rate amount for merchant based on each item.
blockageRateAmountSubMerchant (ItemTransactions)	Decimal	Blockage rate amount for submerchant based on each item.
blockageResolvedDate (ItemTransactions)	String	Blockage resolve date base on item. Date format 2015-10-19 14:36:52
iyziCommissionFee (ItemTransactions)	Decimal	Blockage resolve date base on item. Date format 2015-10-19 14:36:52
iyziCommissionRateAmount (ItemTransactions)	Decimal	Blockage resolve date base on item. Date format 2015-10-19 14:36:52
merchantCommissionRate (ItemTransactions)	Decimal	Üye işyerinin uyguladığı vade/komisyon oranının kırılım bazında dağılmış oranı
merchantCommissionRateAmount (ItemTransactions)	Decimal	E.g. example item price=100, item paidPrice=110 merchant commission rate is %10
subMerchantKey (ItemTransactions)	String	Valid for marketplace merchants.
subMerchantPrice (ItemTransactions)	Decimal	
merchantPayoutAmount (ItemTransactions)	Decimal	payout amount for this item transaction. This is amount of money remaining from item transaction paidPrice after subtracting iyzicoCommissionRateAmount, iyzicoCommissionFee and blockageRateAmountMerchant
subMerchantPayoutAmount (ItemTransactions)	Decimal	Remaining amount after iyzico commissiyon fee and rate amount, blockage amount

subMerchantPayoutRate (ItemTransactions)	Decimal	SubMerchant payout amount rate - It will be %90 if submerchant price is 45 TL and price is 50 TL.
---	---------	---

### 3.3.1.3 Non 3DS Payment

#### 3.3.1.3.1 Payment Process for Product Group, Marketplaces Merchants;

Basket contains 3 different products and basket price is 100 TL. Installment value is 3 and by adding installment charge paid price is 110 TL. One of item in basket is (itemType) physical product that is why shippingAddress have to send within request.

Items are belongs to 3 different submerchants.

- First item's price is 30 TL and subMerchant price is 27 TL (Merchant commission rate is %10).
- Second item's price is 50 TL and subMerchant price is 42 TL (Merchant commission rate is %16).
- Third item's price is 20 TL and subMerchant price is 18 (Merchant commission rate is %10).
- **API Service (Endpoint) Name: /payment/iyzipos/auth/ecom**
- **Method: HTTP POST**

#### Request Example (java):

```
EcomPaymentServiceClient client = new EcomPaymentServiceClientImpl(configuration);

EcomPaymentAuthRequest request = new EcomPaymentAuthRequest();
request.setLocale(RequestLocaleType.TR.getValue());
request.setConversationId("123456789");
request.setPrice(new BigDecimal("100"));
request.setPaidPrice(new BigDecimal("110"));
request.setCurrency ("TRY");
request.setInstallment(3);
request.setBasketId("B67832");
request.setPaymentChannel(PaymentChannelRequestType.WEB.name());
request.setPaymentGroup(PaymentGroupRequestType.PRODUCT.name());
```

```

EcomPaymentCardDto paymentCard = new EcomPaymentCardDto();
paymentCard.setCardHolderName("Sabri Onur Tüzün");
paymentCard.setCardNumber("554960*****");
paymentCard.setExpireMonth("**");
paymentCard.setExpireYear("20**");
paymentCard.setCvc("***");
paymentCard.setRegisterCard(0);
request.setPaymentCard(paymentCard);

EcomPaymentBuyerDto buyer = new EcomPaymentBuyerDto();
buyer.setId("BY789");
buyer.setName("Sabri Onur");
buyer.setSurname("Tüzün");
buyer.setGsmNumber("+905350000000");
buyer.setEmail("email@buyeremail.com");
buyer.setIdentityNumber("74300864791");
buyer.setLastLoginDate("2015-10-05 12:43:35");
buyer.setRegistrationDate("2013-04-21 15:12:09");
buyer.setRegistrationAddress("Nidakule Göztepe İş Merkezi Merdivenköy Mah. Bora Sok. No:1 Kat:19
Bağımsız 70/73 Göztepe Kadıköy");
buyer.setIp("85.34.78.112");
buyer.setCity("İstanbul");
buyer.setCountry("Türkiye");
buyer.setZipCode("34732");
request.setBuyer(buyer);

EcomPaymentShippingAddressDto shippingAddress = new EcomPaymentShippingAddressDto();
shippingAddress.setContactName("Hakan Erdoğan");
shippingAddress.setCity("İstanbul");
shippingAddress.setCountry("Türkiye");
shippingAddress.setAddress("19 Mayıs Mah. İnönü Cad. No:45 Kozyatağı");
shippingAddress.setZipCode("34742");
request.setShippingAddress(shippingAddress);

EcomPaymentBillingAddressDto billingAddress = new EcomPaymentBillingAddressDto();
billingAddress.setContactName("Hakan Erdoğan");
billingAddress.setCity("İstanbul");
billingAddress.setCountry("Türkiye");
billingAddress.setAddress("19 Mayıs Mah. İnönü Cad. No:45 Kozyatağı");
billingAddress.setZipCode("34742");
request.setBillingAddress(billingAddress);

List<EcomPaymentBasketItemDto> basketItems = new ArrayList<EcomPaymentBasketItemDto>();

```

```
EcomPaymentBasketItemDto basketItem1 = new EcomPaymentBasketItemDto();
basketItem1.setId("BI101");
basketItem1.setName("ABC Marka Kolye");
basketItem1.setCategory1("Giyim");
basketItem1.setCategory2("Aksesuar");
basketItem1.setItemType(BasketItemRequestType.PHYSICAL.name());
basketItem1.setPrice(new BigDecimal("30"));
basketItem1.setSubMerchantKey("I4FCFyclof0paTP6687dOoch9Po=");
basketItem1.setSubMerchantPrice(new BigDecimal("27"));
basketItems.add(basketItem1);
```

```
EcomPaymentBasketItemDto basketItem2 = new EcomPaymentBasketItemDto();
basketItem2.setId("BI102");
basketItem2.setName("XYZ Oyun Kodu");
basketItem2.setCategory1("Oyun");
basketItem2.setCategory2("Online Oyun Kodları");
basketItem2.setItemType(BasketItemRequestType.VIRTUAL.name());
basketItem2.setPrice(new BigDecimal("50"));
basketItem2.setSubMerchantKey("IcB6bA9DWEF5e8yGJI0uIEcW1kw=");
basketItem2.setSubMerchantPrice(new BigDecimal("42"));
basketItems.add(basketItem2);
```

```
EcomPaymentBasketItemDto basketItem3 = new EcomPaymentBasketItemDto();
basketItem3.setId("BI103");
basketItem3.setName("EDC Marka Usb");
basketItem3.setCategory1("Elektronik");
basketItem3.setCategory2("Usb / Cable");
basketItem3.setItemType(BasketItemRequestType.PHYSICAL.name());
basketItem3.setPrice(new BigDecimal("20"));
basketItem3.setSubMerchantKey("D9V/MqIRitUzA4dutL+nCBvnWfs=");
basketItem3.setSubMerchantPrice(new BigDecimal("18"));
basketItems.add(basketItem3);
request.setBasketItems(basketItems);
```

```
EcomPaymentAuthResponse response = client.auth(request);
```

```
System.out.println("conversationId=" + response.getConversationId());
System.out.println("locale=" + response.getLocale());
System.out.println("status=" + response.getStatus());
System.out.println("errorCode=" + response.getErrorCode());
System.out.println("errorMessage=" + response.getErrorMessage());
System.out.println("errorGroup=" + response.getErrorGroup());
System.out.println("systemTime=" + response.getSystemTime());
```

```

System.out.println("paymentId=" + response.getPaymentId());
System.out.println("price=" + response.getPrice());
System.out.println("paidPrice=" + response.getPaidPrice());
System.out.println("currency=" + response.getCurrency());
System.out.println("installment=" + response.getInstallment());
System.out.println("basketId=" + response.getBasketId());
System.out.println("binNumber=" + response.getBinNumber());
System.out.println("cardAssociation=" + response.getCardAssociation());
System.out.println("cardFamily=" + response.getCardFamily());
System.out.println("cardType=" + response.getCardType());
System.out.println("cardToken=" + response.getCardToken());
System.out.println("cardUserKey=" + response.getCardUserKey());
System.out.println("fraudStatus=" + response.getFraudStatus());
System.out.println("iyziCommissionFee=" + response.getIyziCommissionFee());
System.out.println("iyziCommissionRateAmount=" + response.getIyziCommissionRateAmount());
System.out.println("merchantCommissionRate=" + response.getMerchantCommissionRate());
System.out.println("merchantCommissionRateAmount=" +
response.getMerchantCommissionRateAmount());

List<EcomPaymentItemTransactionDto> itemTransactions = response.getItemTransactions();

if (itemTransactions != null && !itemTransactions.isEmpty()) {
    for (EcomPaymentItemTransactionDto itemTransaction : itemTransactions) {
        System.out.println("paymentTransactionId=" + itemTransaction.getPaymentTransactionId());
        System.out.println("itemId=" + itemTransaction.getItemId());
        System.out.println("paidPrice=" + itemTransaction.getPaidPrice());
        System.out.println("price=" + itemTransaction.getPrice());
        System.out.println("transactionStatus=" + itemTransaction.getTransactionStatus());
        System.out.println("blockageRate=" + itemTransaction.getBlockageRate());
        System.out.println("blockageRateAmountMerchant=" +
itemTransaction.getBlockageRateAmountMerchant());
        System.out.println("blockageRateAmountSubMerchant=" +
itemTransaction.getBlockageRateAmountSubMerchant());
        System.out.println("blockageResolvedDate=" + itemTransaction.getBlockageResolvedDate());
        System.out.println("iyziCommissionFee=" + itemTransaction.getIyziCommissionFee());
        System.out.println("iyziCommissionRateAmount=" +
itemTransaction.getIyziCommissionRateAmount());
        System.out.println("merchantCommissionRate=" +
itemTransaction.getMerchantCommissionRate());
        System.out.println("merchantCommissionRateAmount=" +
itemTransaction.getMerchantCommissionRateAmount());
        System.out.println("merchantPayoutAmount=" + itemTransaction.getMerchantPayoutAmount());
        System.out.println("subMerchantKey=" + itemTransaction.getSubMerchantKey());
    }
}

```

```

System.out.println("subMerchantPayoutAmount=" +
itemTransaction.getSubMerchantPayoutAmount());
System.out.println("subMerchantPayoutRate=" + itemTransaction.getSubMerchantPayoutRate());
System.out.println("subMerchantPrice=" + itemTransaction.getSubMerchantPrice());
}
}

```

In the example below; there is 3 items in basket, installment is 3, price 100 TL and paidPrice is 110 TL.

#### JSON format of request example:

```

{
  "locale" : "tr",
  "conversationId" : "123456789",
  "price" : 100,
  "paidPrice" : 110,
  "currency" : TRY,
  "installment" : 3,
  "paymentChannel" : "WEB",
  "basketId" : "B67832",
  "paymentGroup" : "PRODUCT",
  "paymentCard" : {
    "cardHolderName" : "Sabri Onur Tüzün",
    "cardNumber" : "554960*****",
    "expireYear" : "20**",
    "expireMonth" : "**",
    "cvc" : "**",
    "registerCard" : 0
  },
  "buyer" : {
    "id" : "BY789",
    "name" : "Sabri Onur",
    "surname" : "Tüzün",
    "identityNumber" : "74300864791",
    "email" : "email@buyeremail.com",
    "gsmNumber" : "+905350000000",
    "registrationDate" : "2013-04-21 15:12:09",
    "lastLoginDate" : "2015-10-05 12:43:35",
    "registrationAddress" : "Nidakule Göztepe İş Merkezi Merdivenköy Mah. Bora Sok. No:1 Kat:19
Bağımsız 70/73 Göztepe Kadıköy",
    "city" : "İstanbul",
    "country" : "Türkiye",

```



```
"zipCode" : "34732",
"ip" : "85.34.78.112"
},
"shippingAddress" : {
  "address" : "19 Mayıs Mah. İnönü Cad. No:45 Kozyatağı",
  "zipCode" : "34742",
  "contactName" : "Hakan Erdoğan",
  "city" : "İstanbul",
  "country" : "Türkiye"
},
"billingAddress" : {
  "address" : "19 Mayıs Mah. İnönü Cad. No:45 Kozyatağı",
  "zipCode" : "34742",
  "contactName" : "Hakan Erdoğan",
  "city" : "İstanbul",
  "country" : "Türkiye"
},
"basketItems" : [ {
  "id" : "BI101",
  "price" : 30,
  "name" : "ABC Marka Kolye",
  "category1" : "Giyim",
  "category2" : "Aksesuar",
  "itemType" : "PHYSICAL",
  "subMerchantKey" : "I4FCFycl0f0paTP6687dOoch9Po=",
  "subMerchantPrice" : 27
}, {
  "id" : "BI102",
  "price" : 50,
  "name" : "XYZ Oyun Kodu",
  "category1" : "Oyun",
  "category2" : "Online Oyun Kodları",
  "itemType" : "VIRTUAL",
  "subMerchantKey" : "IcB6bA9DWEF5e8yGJI0ulEcW1kw=",
  "subMerchantPrice" : 42
}, {
  "id" : "BI103",
  "price" : 20,
  "name" : "EDC Marka Usb",
  "category1" : "Elektronik",
  "category2" : "Usb / Cable",
  "itemType" : "PHYSICAL",
  "subMerchantKey" : "D9V/MqIRitUzA4dutL+nCBvnWfs=",
```

```
"subMerchantPrice" : 18
}]
}
```

#### JSON format of response example:

```
{
  "status" : "success",
  "locale" : "tr",
  "systemTime" : 1444142311338,
  "conversationId" : "123456789",
  "price" : 100,
  "paidPrice" : 110,
  "currency" : TRY,
  "installment" : 3,
  "paymentId" : "2744",
  "fraudStatus" : 1,
  "merchantCommissionRate" : 10.00000000,
  "merchantCommissionRateAmount" : 10,
  "iyziCommissionRateAmount" : 4.51000000,
  "iyziCommissionFee" : 0.25000000,
  "cardType" : "CREDIT_CARD",
  "cardAssociation" : "MASTER_CARD",
  "cardFamily" : "Bonus",
  "binNumber" : "554960",
  "basketId" : "B67832",
  "itemTransactions" : [ {
    "itemId" : "BI101",
    "paymentTransactionId" : "1307",
    "transactionStatus" : 1,
    "price" : 30,
    "paidPrice" : 33.00000000,
    "merchantCommissionRate" : 10.00000000,
    "merchantCommissionRateAmount" : 3.00000000,
    "iyziCommissionRateAmount" : 1.35300000,
    "iyziCommissionFee" : 0.07500000,
    "blockageRate" : 7.00000000,
    "blockageRateAmountMerchant" : 2.31000000,
    "blockageRateAmountSubMerchant" : 0,
    "blockageResolvedDate" : "2015-10-16 17:38:31",
    "subMerchantKey" : "l4FCFycl0f0paTP6687dOoch9Po=",
    "subMerchantPrice" : 27,
    "subMerchantPayoutRate" : 90.00000000,
```

```

"subMerchantPayoutAmount" : 27,
"merchantPayoutAmount" : 2.26200000
}, {
  "itemId" : "BI102",
  "paymentTransactionId" : "1308",
  "transactionStatus" : 1,
  "price" : 50,
  "paidPrice" : 55.00000000,
  "merchantCommissionRate" : 10.00000000,
  "merchantCommissionRateAmount" : 5.00000000,
  "iyziCommissionRateAmount" : 2.25500000,
  "iyziCommissionFee" : 0.12500000,
  "blockageRate" : 7.00000000,
  "blockageRateAmountMerchant" : 3.85000000,
  "blockageRateAmountSubMerchant" : 0,
  "blockageResolvedDate" : "2015-10-16 17:38:31",
  "subMerchantKey" : "IcB6bA9DWEF5e8yGJI0uIEcW1kw=",
  "subMerchantPrice" : 42,
  "subMerchantPayoutRate" : 84.00000000,
  "subMerchantPayoutAmount" : 42,
  "merchantPayoutAmount" : 6.77000000
}, {
  "itemId" : "BI103",
  "paymentTransactionId" : "1309",
  "transactionStatus" : 1,
  "price" : 20,
  "paidPrice" : 22.00000000,
  "merchantCommissionRate" : 10.00000000,
  "merchantCommissionRateAmount" : 2.00000000,
  "iyziCommissionRateAmount" : 0.90200000,
  "iyziCommissionFee" : 0.05000000,
  "blockageRate" : 7.00000000,
  "blockageRateAmountMerchant" : 1.54000000,
  "blockageRateAmountSubMerchant" : 0,
  "blockageResolvedDate" : "2015-10-16 17:38:31",
  "subMerchantKey" : "D9V/MqIRitUzA4dutL+nCBvnWfs=",
  "subMerchantPrice" : 18,
  "subMerchantPayoutRate" : 90.00000000,
  "subMerchantPayoutAmount" : 18,
  "merchantPayoutAmount" : 1.50800000
}]
}

```

### 3.3.1.3.2 Payment Process of LISTING and SUBSCRIPTION (Same process for marketplace merchants and non-marketplace merchants);

Merchants may have payments for SUBSCRIPTION and LISTING paymentGroups regardless to its business model.

In a marketplace business model there is payment process for SUBSCRIPTION from subMerchant to merchant.

For example, there is 3 item in the basket and all item's paymentGroup is SUBSCRIPTION. Total price is 100 TL, installment value is 3 and paidprice is 110 TL. Carrier process is essential for one of the item even its productGroup is SUBSCRIPTION, so productType parameter should be PHYSICAL. Therefor shippingAddress parameter also send within request.

- API Service (Endpoint) Name: `/payment/iyzipos/auth/ecom`
- Method: **HTTP POST**

#### Request Example (java):

```
EcomPaymentServiceClient client = new EcomPaymentServiceClientImpl(configuration);

EcomPaymentAuthRequest request = new EcomPaymentAuthRequest();
request.setLocale(RequestLocaleType.TR.getValue());
request.setConversationId("123456789");
request.setPrice(new BigDecimal("100"));
request.setPaidPrice(new BigDecimal("110"));
request.setCurrency ("TRY");

request.setInstallment(3);
request.setBasketId("B67832");
request.setPaymentChannel(PaymentChannelRequestType.WEB.name());
request.setPaymentGroup(PaymentGroupRequestType.SUBSCRIPTION.name());

EcomPaymentCardDto paymentCard = new EcomPaymentCardDto();
paymentCard.setCardHolderName("Sabri Onur Tüzün");
paymentCard.setCardNumber("554960*****7015");
```

```

paymentCard.setExpireMonth("**");
paymentCard.setExpireYear("20**");
paymentCard.setCvc("***");
paymentCard.setRegisterCard(0);
request.setPaymentCard(paymentCard);

EcomPaymentBuyerDto buyer = new EcomPaymentBuyerDto();
buyer.setId("BY789");
buyer.setName("Sabri Onur");
buyer.setSurname("Tüzün");
buyer.setGsmNumber("+905350000000");
buyer.setEmail("onur.tuzun@iyzico.com");
buyer.setIdentityNumber("74300864791");
buyer.setLastLoginDate("2015-10-05 12:43:35");
buyer.setRegistrationDate("2013-04-21 15:12:09");
buyer.setRegistrationAddress("Nidakule Göztepe İş Merkezi Merdivenköy Mah. Bora Sok. No:1 Kat:19
Bağımsız 70/73 Göztepe Kadıköy");
buyer.setIp("85.34.78.112");
buyer.setCity("İstanbul");
buyer.setCountry("Türkiye");
buyer.setZipCode("34732");
request.setBuyer(buyer);

EcomPaymentShippingAddressDto shippingAddress = new EcomPaymentShippingAddressDto();
shippingAddress.setContactName("Hakan Erdoğan");
shippingAddress.setCity("İstanbul");
shippingAddress.setCountry("Türkiye");
shippingAddress.setAddress("19 Mayıs Mah. İnönü Cad. No:45 Kozyatağı");
shippingAddress.setZipCode("34742");
request.setShippingAddress(shippingAddress);

EcomPaymentBillingAddressDto billingAddress = new EcomPaymentBillingAddressDto();
billingAddress.setContactName("Hakan Erdoğan");
billingAddress.setCity("İstanbul");
billingAddress.setCountry("Türkiye");
billingAddress.setAddress("19 Mayıs Mah. İnönü Cad. No:45 Kozyatağı");
billingAddress.setZipCode("34742");
request.setBillingAddress(billingAddress);

List<EcomPaymentBasketItemDto> basketItems = new ArrayList<EcomPaymentBasketItemDto>();
EcomPaymentBasketItemDto basketItem1 = new EcomPaymentBasketItemDto();
basketItem1.setId("BI101");
basketItem1.setName("Dükkan aboneliği ve katalog");

```

```

basketItem1.setCategory1("Abonelik");
basketItem1.setCategory2("Dükkan");
basketItem1.setItemType(BasketItemRequestType.PHYSICAL.name());
basketItem1.setPrice(new BigDecimal("30"));
basketItems.add(basketItem1);

EcomPaymentBasketItemDto basketItem2 = new EcomPaymentBasketItemDto();
basketItem2.setId("BI102");
basketItem2.setName("Listeleme aboneliği");
basketItem2.setCategory1("Abonelik");
basketItem2.setCategory2("Listeleme");
basketItem2.setItemType(BasketItemRequestType.VIRTUAL.name());
basketItem2.setPrice(new BigDecimal("50"));
basketItems.add(basketItem2);

EcomPaymentBasketItemDto basketItem3 = new EcomPaymentBasketItemDto();
basketItem3.setId("BI103");
basketItem3.setName("Servis aboneliği");
basketItem3.setCategory1("Abonelik");
basketItem3.setCategory2("Servis");
basketItem3.setItemType(BasketItemRequestType.VIRTUAL.name());
basketItem3.setPrice(new BigDecimal("20"));
basketItems.add(basketItem3);
request.setBasketItems(basketItems);

EcomPaymentAuthResponse response = client.auth(request);

System.out.println("conversationId=" + response.getConversationId());
System.out.println("locale=" + response.getLocale());
System.out.println("status=" + response.getStatus());
System.out.println("errorCode=" + response.getErrorCode());
System.out.println("errorMessage=" + response.getErrorMessage());
System.out.println("errorGroup=" + response.getErrorGroup());
System.out.println("systemTime=" + response.getSystemTime());
System.out.println("paymentId=" + response.getPaymentId());
System.out.println("price=" + response.getPrice());
System.out.println("paidPrice=" + response.getPaidPrice());
System.out.println("currency=" + response.getCurrency());
System.out.println("installment=" + response.getInstallment());
System.out.println("basketId=" + response.getBasketId());
System.out.println("binNumber=" + response.getBinNumber());
System.out.println("cardAssociation=" + response.getCardAssociation());
System.out.println("cardFamily=" + response.getCardFamily());

```

```

System.out.println("cardType=" + response.getCardType());
System.out.println("cardToken=" + response.getCardToken());
System.out.println("cardUserKey=" + response.getCardUserKey());
System.out.println("fraudStatus=" + response.getFraudStatus());
System.out.println("iyziCommissionFee=" + response.getIyziCommissionFee());
System.out.println("iyziCommissionRateAmount=" + response.getIyziCommissionRateAmount());
System.out.println("merchantCommissionRate=" + response.getMerchantCommissionRate());
System.out.println("merchantCommissionRateAmount=" +
response.getMerchantCommissionRateAmount());

List<EcomPaymentItemTransactionDto> itemTransactions = response.getItemTransactions();

if (itemTransactions != null && !itemTransactions.isEmpty()) {
    for (EcomPaymentItemTransactionDto itemTransaction : itemTransactions) {
        System.out.println("paymentTransactionId=" + itemTransaction.getPaymentTransactionId());
        System.out.println("itemId=" + itemTransaction.getItemId());
        System.out.println("paidPrice=" + itemTransaction.getPaidPrice());
        System.out.println("price=" + itemTransaction.getPrice());
        System.out.println("transactionStatus=" + itemTransaction.getTransactionStatus());
        System.out.println("blockageRate=" + itemTransaction.getBlockageRate());
        System.out.println("blockageRateAmountMerchant=" +
itemTransaction.getBlockageRateAmountMerchant());
        System.out.println("blockageRateAmountSubMerchant=" +
itemTransaction.getBlockageRateAmountSubMerchant());
        System.out.println("blockageResolvedDate=" + itemTransaction.getBlockageResolvedDate());
        System.out.println("iyziCommissionFee=" + itemTransaction.getIyziCommissionFee());
        System.out.println("iyziCommissionRateAmount=" +
itemTransaction.getIyziCommissionRateAmount());
        System.out.println("merchantCommissionRate=" +
itemTransaction.getMerchantCommissionRate());
        System.out.println("merchantCommissionRateAmount=" +
itemTransaction.getMerchantCommissionRateAmount());
        System.out.println("merchantPayoutAmount=" + itemTransaction.getMerchantPayoutAmount());
    }
}

```

### Response Example (java):

```

conversationId=123456789
locale=tr
status=success
errorCode=null

```

```

errorMessage=null
errorGroup=null
systemTime=1444735597648
paymentId=667906
price=100
paidPrice=110
currency= TRY
installment=3
basketId=B67832
binNumber=554960
cardAssociation=MASTER_CARD
cardFamily=Bonus
cardType=CREDIT_CARD
cardToken=null
cardUserKey=null
fraudStatus=1
iyziCommissionFee=0.25000000
iyziCommissionRateAmount=4.51000000
merchantCommissionRate=10.00000000
merchantCommissionRateAmount=10
paymentTransactionId=506591
itemId=BI101
paidPrice=33.00000000
price=30
transactionStatus=2
blockageRate=10.00000000
blockageRateAmountMerchant=3.30000000
blockageRateAmountSubMerchant=0
blockageResolvedDate=2015-10-28 14:26:37
iyziCommissionFee=0.07500000
iyziCommissionRateAmount=1.35300000
merchantCommissionRate=10.00000000
merchantCommissionRateAmount=3.00000000
merchantPayoutAmount=28.27200000
paymentTransactionId=506596
itemId=BI102
paidPrice=55.00000000
price=50
transactionStatus=2
blockageRate=10.00000000
blockageRateAmountMerchant=5.50000000
blockageRateAmountSubMerchant=0
blockageResolvedDate=2015-10-28 14:26:37

```



```

iyziCommissionFee=0.12500000
iyziCommissionRateAmount=2.25500000
merchantCommissionRate=10.00000000
merchantCommissionRateAmount=5.00000000
merchantPayoutAmount=47.12000000
paymentTransactionId=506601
itemId=BI103
paidPrice=22.00000000
price=20
transactionStatus=2
blockageRate=10.00000000
blockageRateAmountMerchant=2.20000000
blockageRateAmountSubMerchant=0
blockageResolvedDate=2015-10-28 14:26:37
iyziCommissionFee=0.05000000
iyziCommissionRateAmount=0.90200000
merchantCommissionRate=10.00000000
merchantCommissionRateAmount=2.00000000
merchantPayoutAmount=18.84800000

```

#### JSON format of request example:

```

{
  "locale" : "tr",
  "conversationId" : "123456789",
  "price" : 100,
  "paidPrice" : 110,
  "currency" : TRY,
  "installment" : 3,
  "paymentChannel" : "WEB",
  "basketId" : "B67832",
  "paymentGroup" : "SUBSCRIPTION",
  "paymentCard" : {
    "cardHolderName" : "Sabri Onur Tüzün",
    "cardNumber" : "554960*****7015",
    "expireYear" : "20**",
    "expireMonth" : "***",
    "cvc" : "****",
    "registerCard" : 0
  },
  "buyer" : {
    "id" : "BY789",

```

```

"name" : "Sabri Onur",
"surname" : "Tüzün",
"identityNumber" : "74300864791",
"email" : "onur.tuzun@iyzico.com",
"gsmNumber" : "+905350000000",
"registrationDate" : "2013-04-21 15:12:09",
"lastLoginDate" : "2015-10-05 12:43:35",
"registrationAddress" : "Nidakule Göztepe İş Merkezi Merdivenköy Mah. Bora Sok. No:1 Kat:19
Bağımsız 70/73 Göztepe Kadıköy",
"city" : "İstanbul",
"country" : "Türkiye",
"zipCode" : "34732",
"ip" : "85.34.78.112"
},
"shippingAddress" : {
"address" : "19 Mayıs Mah. İnönü Cad. No:45 Kozyatağı",
"zipCode" : "34742",
"contactName" : "Hakan Erdoğan",
"city" : "İstanbul",
"country" : "Türkiye"
},
"billingAddress" : {
"address" : "19 Mayıs Mah. İnönü Cad. No:45 Kozyatağı",
"zipCode" : "34742",
"contactName" : "Hakan Erdoğan",
"city" : "İstanbul",
"country" : "Türkiye"
},
"basketItems" : [ {
"id" : "BI101",
"price" : 30,
"name" : "Dükkan aboneliği ve katalog",
"category1" : "Abonelik",
"category2" : "Dükkan",
"itemType" : "PHYSICAL"
}, {
"id" : "BI102",
"price" : 50,
"name" : "Listeleme aboneliği",
"category1" : "Abonelik",
"category2" : "Listeleme",
"itemType" : "VIRTUAL"
}, {

```

```

    "id" : "BI103",
    "price" : 20,
    "name" : "Servis aboneliği",
    "category1" : "Abonelik",
    "category2" : "Servis",
    "itemType" : "VIRTUAL"
  }
}

```

#### JSON format of response example :

```

{
  "status" : "success",
  "locale" : "tr",
  "systemTime" : 1444735597648,
  "conversationId" : "123456789",
  "price" : 100,
  "paidPrice" : 110,
  "currency" : TRY,
  "installment" : 3,
  "paymentId" : "667906",
  "fraudStatus" : 1,
  "merchantCommissionRate" : 10.00000000,
  "merchantCommissionRateAmount" : 10,
  "iyziCommissionRateAmount" : 4.51000000,
  "iyziCommissionFee" : 0.25000000,
  "cardType" : "CREDIT_CARD",
  "cardAssociation" : "MASTER_CARD",
  "cardFamily" : "Bonus",
  "binNumber" : "554960",
  "basketId" : "B67832",
  "itemTransactions" : [ {
    "itemId" : "BI101",
    "paymentTransactionId" : "506591",
    "transactionStatus" : 2,
    "price" : 30,
    "paidPrice" : 33.00000000,
    "merchantCommissionRate" : 10.00000000,
    "merchantCommissionRateAmount" : 3.00000000,
    "iyziCommissionRateAmount" : 1.35300000,
    "iyziCommissionFee" : 0.07500000,
    "blockageRate" : 10.00000000,
    "blockageRateAmountMerchant" : 3.30000000,

```

```

"blockageRateAmountSubMerchant" : 0,
"blockageResolvedDate" : "2015-10-28 14:26:37",
"subMerchantPrice" : 0,
"subMerchantPayoutRate" : 0E-8,
"subMerchantPayoutAmount" : 0,
"merchantPayoutAmount" : 28.27200000
}, {
"itemId" : "BI102",
"paymentTransactionId" : "506596",
"transactionStatus" : 2,
"price" : 50,
"paidPrice" : 55.00000000,
"merchantCommissionRate" : 10.00000000,
"merchantCommissionRateAmount" : 5.00000000,
"iyziCommissionRateAmount" : 2.25500000,
"iyziCommissionFee" : 0.12500000,
"blockageRate" : 10.00000000,
"blockageRateAmountMerchant" : 5.50000000,
"blockageRateAmountSubMerchant" : 0,
"blockageResolvedDate" : "2015-10-28 14:26:37",
"subMerchantPrice" : 0,
"subMerchantPayoutRate" : 0E-8,
"subMerchantPayoutAmount" : 0,
"merchantPayoutAmount" : 47.12000000
}, {
"itemId" : "BI103",
"paymentTransactionId" : "506601",
"transactionStatus" : 2,
"price" : 20,
"paidPrice" : 22.00000000,
"merchantCommissionRate" : 10.00000000,
"merchantCommissionRateAmount" : 2.00000000,
"iyziCommissionRateAmount" : 0.90200000,
"iyziCommissionFee" : 0.05000000,
"blockageRate" : 10.00000000,
"blockageRateAmountMerchant" : 2.20000000,
"blockageRateAmountSubMerchant" : 0,
"blockageResolvedDate" : "2015-10-28 14:26:37",
"subMerchantPrice" : 0,
"subMerchantPayoutRate" : 0E-8,
"subMerchantPayoutAmount" : 0,
"merchantPayoutAmount" : 18.84800000
}]

```

```
}
```

### 3.3.1.3.3 Payment Process of Product, Non-Marketplae Merchants;

It is almoslty same process with LISTING and SUBSCRIPTION productGroups. Differences,

- productType parameter must set as PHYSICAL and shippingAddress parameter should send within request.
- subMerchantKey and subMerchantPrice are not applicable.
- paymentGroup parameter must set as PRODUCT within request.

```
request.setPaymentGroup(PaymentGroupRequestType.PRODUCT.name());
```

### 3.3.1.4 3D Secure Payment

Merchants may prefer 3D Secure payment regarding to many conditions such as fraud risk, debit card, buyer choice.

3D Secure payment process has consist of 3 steps.

1. **initialize 3DS** : Basket details, buyer details and callback URL are required parameters for initialize3DS request. iyzico will return “threeDSHtmlContent” parameter within response. If the status of response is success, merchants should append “threeDSHtmlContent” value to browser’s response body. After merchants append “threeDSHtmlContent” value, buyer will redirect to 3DS payment page.

**Note:** iyzico return HTML content encoded with Base64 and iyzico clients able to decode it. So merchants only see user friendly content.

2. **Callback** : Buyer entire OTP (one time password) which sent by issuer bank, into 3DS payment page to authanticate payments. iyzico recieve response for OTP authantication from issuer bank. Then iyzico apply its own validations. After that iyzico send HTML content to merchant’s callback URL. Merhants apply their own validation such as stock control, product avaliability and etc. Now there will be a handshake process between 3

parties; issuer bank, Merchant and iyzico. At this point payment doesn't collect from buyer account.

- 3. Auth 3DS :** Status of response is shown in HTML content. If status of response is success merchants will apply their own validations. Then merchants send *auth3DS* request to make collection. If status of response is fail, *auth3DS* request shouldn't send.

#### 3.3.1.4.1 3D Secure Payment Process for Product Group, Marketplaces Merchants

There is a basket consist of 3 different items, total amount of basket (basketPrice) is 100 TL.

Installment value is 3 and paid price is 110 TL. Payment type is 3D Secure.

One of item's productType in the basket PHYSICAL therefore shippingAddress parameter should send within payment request.

- First item's price is 30 TL and after item approval submerchantPrice is 27 TL. (merchantCommissionRate is %10)
- Second item's price is 50 TL and after item approval submerchantPrice is 42 TL. (merchantCommissionRate is %16)
- Third item's price is 20 TL and after item approval submerchantPrice is 18 TL. (merchantCommissionRate is %10)

#### **initialize3DS for Product Group:**

- API Service (Endpoint) Name: **/payment/iyzipos/initialize3ds/ecom**
- Method: **HTTP POST**

#### **Example of initialize3DS Request:**

```
EcomPaymentServiceClient client = new EcomPaymentServiceClientImpl(configuration);
```

```
EcomPaymentThreeDSInitializeRequest request = new EcomPaymentThreeDSInitializeRequest();
```

```

request.setLocale(RequestLocaleType.TR.getValue());
request.setConversationId("123456789");
request.setCallbackUrl("https://www.merchant.com/callbackUrl");
request.setPrice(new BigDecimal("100"));
request.setPaidPrice(new BigDecimal("110"));
request.setCurrency ("TRY");
request.setInstallment(3);
request.setBasketId("B67832");
request.setPaymentChannel(PaymentChannelRequestType.WEB.name());
request.setPaymentGroup(PaymentGroupRequestType.PRODUCT.name());

```

```

EcomPaymentCardDto paymentCard = new EcomPaymentCardDto();
paymentCard.setCardHolderName("Sabri Onur Tüzün");
paymentCard.setCardNumber("554960****7015");
paymentCard.setExpireMonth("***");
paymentCard.setExpireYear("20**");
paymentCard.setCvc("***");
paymentCard.setRegisterCard(0);
request.setPaymentCard(paymentCard);

```

```

EcomPaymentBuyerDto buyer = new EcomPaymentBuyerDto();
buyer.setId("BY789");
buyer.setName("Sabri Onur");
buyer.setSurname("Tüzün");
buyer.setGsmNumber("+905350000000");
buyer.setEmail("onur.tuzun@iyzico.com");
buyer.setIdentityNumber("74300864791");
buyer.setLastLoginDate("2015-10-05 12:43:35");
buyer.setRegistrationDate("2013-04-21 15:12:09");
buyer.setRegistrationAddress("Nidakule Göztepe İş Merkezi Merdivenköy Mah. Bora Sok. No:1 Kat:19
Bağımsız 70/73 Göztepe Kadıköy");
buyer.setIp("85.34.78.112");
buyer.setCity("İstanbul");
buyer.setCountry("Türkiye");
buyer.setZipCode("34732");
request.setBuyer(buyer);

```

```

EcomPaymentShippingAddressDto shippingAddress = new EcomPaymentShippingAddressDto();
shippingAddress.setContactName("Hakan Erdoğan");
shippingAddress.setCity("İstanbul");
shippingAddress.setCountry("Türkiye");
shippingAddress.setAddress("19 Mayıs Mah. İnönü Cad. No:45 Kozyatağı");
shippingAddress.setZipCode("34742");

```

```
request.setShippingAddress(shippingAddress);
```

```
EcomPaymentBillingAddressDto billingAddress = new EcomPaymentBillingAddressDto();
billingAddress.setContactName("Hakan Erdoğan");
billingAddress.setCity("İstanbul");
billingAddress.setCountry("Türkiye");
billingAddress.setAddress("19 Mayıs Mah. İnönü Cad. No:45 Kozyatağı");
billingAddress.setZipCode("34742");
request.setBillingAddress(billingAddress);
```

```
List<EcomPaymentBasketItemDto> basketItems = new ArrayList<EcomPaymentBasketItemDto>();
EcomPaymentBasketItemDto basketItem1 = new EcomPaymentBasketItemDto();
basketItem1.setId("BI101");
basketItem1.setName("ABC Marka Kolye");
basketItem1.setCategory1("Giyim");
basketItem1.setCategory2("Aksesuar");
basketItem1.setItemType(BasketItemRequestType.PHYSICAL.name());
basketItem1.setPrice(new BigDecimal("30"));
basketItem1.setSubMerchantKey("ha3us4v5mk2652kkjk5728cc4407an");
basketItem1.setSubMerchantPrice(new BigDecimal("27"));
basketItems.add(basketItem1);
```

```
EcomPaymentBasketItemDto basketItem2 = new EcomPaymentBasketItemDto();
basketItem2.setId("BI102");
basketItem2.setName("XYZ Oyun Kodu");
basketItem2.setCategory1("Oyun");
basketItem2.setCategory2("Online Oyun Kodları");
basketItem2.setItemType(BasketItemRequestType.VIRTUAL.name());
basketItem2.setPrice(new BigDecimal("50"));
basketItem2.setSubMerchantKey("ha3us4v5mk2652kkjk5728cc4407an");
basketItem2.setSubMerchantPrice(new BigDecimal("42"));
basketItems.add(basketItem2);
```

```
EcomPaymentBasketItemDto basketItem3 = new EcomPaymentBasketItemDto();
basketItem3.setId("BI103");
basketItem3.setName("EDC Marka Usb");
basketItem3.setCategory1("Elektronik");
basketItem3.setCategory2("Usb / Cable");
basketItem3.setItemType(BasketItemRequestType.PHYSICAL.name());
basketItem3.setPrice(new BigDecimal("20"));
basketItem3.setSubMerchantKey("ha3us4v5mk2652kkjk5728cc4407an");
basketItem3.setSubMerchantPrice(new BigDecimal("18"));
basketItems.add(basketItem3);
```



```
request.setBasketItems(basketItems);

EcomPaymentThreeDSInitializeResponse response = client.initializeThreeDS(request);

System.out.println("conversationId=" + response.getConversationId());
System.out.println("locale=" + response.getLocale());
System.out.println("status=" + response.getStatus());
System.out.println("errorCode=" + response.getErrorCode());
System.out.println("errorMessage=" + response.getErrorMessage());
System.out.println("errorGroup=" + response.getErrorGroup());
System.out.println("systemTime=" + response.getSystemTime());
System.out.println("threeDSHtmlContent=" + response.getThreeDSHtmlContent());
```

### Example of initialize3DS Response:

```
conversationId=123456789
locale=tr
status=success
errorCode=null
errorMessage=null
errorGroup=null
systemTime=1444739797682
threeDSHtmlContent=<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html>
<head>
<title>MDpay default response template for web</title>
</head>
<body bgcolor="#02014E" OnLoad="OnLoadEvent();" >
<form name="downloadForm"
action="https://3dsecure.garanti.com.tr/acs/0036537396382857515710002306/merchant/control/master"
method="POST">
<input type="hidden"
name="PaReq"
value="eJxVUdtuwjAM/ZWKD2iSlvSCTCQuD0MaE4Jp0vZWUg860RTSdIV9/ZKWY/YUn2PHxz6G17
1GnG9QNhoFLLGusx16RT4e8GESM87ZQMBqssaTgG/UdVEpwXzqB0Bu0P7Scp8plyCTp+niRfCARjwEco
VQol7MRcSjiCYRS4H0BKisRFFcfgpZAekAyKpRRi9EnFqFG4BGH8TemOOIkLZt/f6LL6sSiEsBeYywalxU21b
nIhdbs522qkZy+pzVanZ+xo/IG8OvzWE3BulqIM8MioAyzigLPRaMhnQUWvGOH6x0M4jX9bvHGpUptVv
1FByd0qQHNudSfymwImpU8iLSov35jgDPx0qhrbAi9xhyrKXwrKx7gTzWmD05X6WxhIEaWI/jMI3CJEh4z
BmPnXAQ0si53RW5/oU1jSWMdQIOAHFtyPWQ5HpnG/27/y9Ltqri">
```

```

<input type="hidden"
  name="TermUrl" value="https://spos.isbank.com.tr/fim/est3Dgate?msgid=54871551">
<input type="hidden"
  name="MD"
  value="554960:49B5838C731576E72B2BB10319D348D301545852C82BB0A45A0E9B11ECB86CE7
:3668:##700656608619">
<!-- To support javascript unaware/disabled browsers -->
<noscript>
  <center>Please click the submit button below.<br>
  <input type="submit" name="submit" value="Submit"></center>
</noscript>
</form>

<SCRIPT LANGUAGE="Javascript" >
<!-- about:blank -->
<!--
function OnLoadEvent() {
  document.downloadForm.submit();
}
//-->
</SCRIPT>
</body>
</html>

```

#### JSON format of initialize3DS request example :

```

{
  "locale" : "tr",
  "conversationId" : "123456789",
  "price" : 100,
  "paidPrice" : 110,
  "currency" : TRY,
  "installment" : 3,
  "paymentChannel" : "WEB",
  "basketId" : "B67832",
  "paymentGroup" : "PRODUCT",
  "paymentCard" : {
    "cardHolderName" : "Sabri Onur Tüzün",
    "cardNumber" : "554960*****7015",
    "expireYear" : "20**",
    "expireMonth" : "**",
    "cvc" : "****",

```

```

"registerCard" : 0
},
"buyer" : {
  "id" : "BY789",
  "name" : "Sabri Onur",
  "surname" : "Tüzün",
  "identityNumber" : "74300864791",
  "email" : "onur.tuzun@iyzico.com",
  "gsmNumber" : "+905350000000",
  "registrationDate" : "2013-04-21 15:12:09",
  "lastLoginDate" : "2015-10-05 12:43:35",
  "registrationAddress" : "Nidakule Göztepe İş Merkezi Merdivenköy Mah. Bora Sok. No:1 Kat:19
Bağımsız 70/73 Göztepe Kadıköy",
  "city" : "İstanbul",
  "country" : "Türkiye",
  "zipCode" : "34732",
  "ip" : "85.34.78.112"
},
"shippingAddress" : {
  "address" : "19 Mayıs Mah. İnönü Cad. No:45 Kozyatağı",
  "zipCode" : "34742",
  "contactName" : "Hakan Erdoğan",
  "city" : "İstanbul",
  "country" : "Türkiye"
},
"billingAddress" : {
  "address" : "19 Mayıs Mah. İnönü Cad. No:45 Kozyatağı",
  "zipCode" : "34742",
  "contactName" : "Hakan Erdoğan",
  "city" : "İstanbul",
  "country" : "Türkiye"
},
"basketItems" : [ {
  "id" : "BI101",
  "price" : 30,
  "name" : "ABC Marka Kolye",
  "category1" : "Giyim",
  "category2" : "Aksesuar",
  "itemType" : "PHYSICAL",
  "subMerchantKey" : "ha3us4v5mk2652kkjk5728cc4407an",
  "subMerchantPrice" : 27
}, {
  "id" : "BI102",

```

```

"price" : 50,
"name" : "XYZ Oyun Kodu",
"category1" : "Oyun",
"category2" : "Online Oyun Kodlari",
"itemType" : "VIRTUAL",
"subMerchantKey" : "ha3us4v5mk2652kkjk5728cc4407an",
"subMerchantPrice" : 42
}, {
"id" : "BI103",
"price" : 20,
"name" : "EDC Marka Usb",
"category1" : "Elektronik",
"category2" : "Usb / Cable",
"itemType" : "PHYSICAL",
"subMerchantKey" : "ha3us4v5mk2652kkjk5728cc4407an",
"subMerchantPrice" : 18
}],
"callbackUrl" : "https://www.merchant.com/callbackUrl"
}

```

#### JSON format of initialize3DS response example :

```

{
  "status" : "success",
  "locale" : "tr",
  "systemTime" : 1444740039787,
  "conversationId" : "123456789",
  "threeDSHtmlContent" : "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN\"
  \">http://www.w3.org/TR/html4/strict.dtd>\n\n<html>\n<head>\n
  <title>MDpay default response
  template for web</title>\n</head>\n<body bgcolor=#02014E\" OnLoad=OnLoadEvent();\n
  >\n<form
  name=downloadForm\" \n action=https://3dsecure.garanti.com.tr/acs/0036537398804063853260059907/merchant/control/master\" \n method=POST\">\n <input
  type=hidden\" \n name=PaReq\" \n value=eJxVUdtuwjAM/ZWKD2iStukFGUsMmECCCW
  19YI9Vao1qawtpu8K+fgnlsj3F59jxsY8h3Wui+RupThPChpom+yCnyCcjGcSRJ4UciWynr3RE+CbdFHWfw
  uWuB+wGzS+t9InVimTq+LR6QenxUPrArhBK0qs5hJlMeRyKBNhAQJWVhMX5p1A1sAsAVXdVq88YUbh
  BqDTX7hv28OYsb7v3eGLq+oSmE0Be4yw7WzUmFanlseMynT9uTs2i3TXr5+XvVgFCzFT6UZNgNkKyLO
  W0ONCCi58R3jjlBj7AbALD1lpZ8D09d0Rgrucm60GCg5WaToAk7OpvxQYSzVV6oxJYFa+I6DToa7IVJgN7z
  Hk1Ch0jKx9gT3WmC2tr6o1hnHuG18jP4ljHvDQj6XvhZzLJOGRdftSZPsXxjQRC3ERsACYbcOuh2TXO5vo3
  /1/AU1fqgg=>\n <input type=hidden\" \n name=TermUrl\"
  value=https://spos.isbank.com.tr/fim/est3Dgate?msgid=54872515\">\n <input

```

```

type="hidden"\n      name="MD"\n      value="554960:4C296BFD8B09CB33B583B7401731DE2
B57E8A0753CE88B889A628E6772B1325B:3867:##700656608619">\n <!-- To support javascript
unaware/disabled browsers -->\n <noscript>\n <center>Please click the submit button
below.<br>\n <input type="submit" name="submit"
value="Submit"></center>\n </noscript>\n</form>\n\n<SCRIPT LANGUAGE="Javascript" >\n <!--
about:blank -->\n <!--\n function OnLoadEvent() {\n document.downloadForm.submit();\n }\n //--
>\n</SCRIPT>\n</body>\n</html>\u0000"
}

```

### Redirect to Merchant (*callback*):

Buyer enter OTP into 3DS payment page and submit it, then iyzico receive response from issuer bank. After iyzico applied its own validation, redirect buyer to merchant's callback URL in browser level.

3 main parameter return in Callback URL (<https://www.merchant.com/callbackUrl>).

- **status** parameter of response (*success* veya *failure*),
- **paymentId** refers to unique id of payment,
- **conversationData** value which is used in data communication.

If *status* of response is *success*, **paymentId** and **conversationData** will send within *auth3DS* request to complete 3D Secure payment process. If *status* of response is *failure*, **paymentId** and **conversationData** parameters will be empty so *auth3DS* request shouldn't send.

### Example of Merchant Redirect HTML Content:

```

<!doctype html>
<html lang="en">
<head>
  <title>Merchant 3ds Callback Form</title>
</head>
<body onload="submitIyzicoThreeDSCallbackForm();">

```

```

<form id="iyzico-3ds-callback-form" name="iyzico-3ds-callback-form"
action="https://www.merchant.com/callbackUrl" method="post">
  <input type="hidden" name="status" value="success">
  <input type="hidden" name="paymentId" value="457932">
  <input type="hidden" name="conversationData" value="KPeA/vSq2nspTXa3mIHveg==">
</form>
<script language="Javascript">
  <!--
  function submitIyzicoThreeDSCallbackForm() {
    document.getElementById("iyzico-3ds-callback-form").submit();
  }
  //-->
</script>
</body>
</html>

```

### **auth3DS :**

After that iyzico send HTML content to merchant's callback URL. Merchants apply their own validation such as stock control, product availability and etc. Then merchants send *auth3DS* request to complete payment.

- **API Service (Endpoint) Name:** `/payment/iyzipos/auth3ds/ecom`
- **Method:** **HTTP POST**

### **Example of Auth3DS Request:**

```

EcomPaymentServiceClient client = new EcomPaymentServiceClientImpl(configuration);

EcomPaymentThreeDSRequest request = new EcomPaymentThreeDSRequest();
request.setLocale(RequestLocaleType.TR.getValue());
request.setConversationId("123456789");
request.setPaymentId("457932");
request.setConversationData("KPeA/vSq2nspTXa3mIHveg==");

EcomPaymentThreeDSResponse response = client.threeDSAuth(request);

System.out.println("conversationId=" + response.getConversationId());
System.out.println("locale=" + response.getLocale());
System.out.println("status=" + response.getStatus());

```

```

System.out.println("errorCode=" + response.getErrorCode());
System.out.println("errorMessage=" + response.getErrorMessage());
System.out.println("errorGroup=" + response.getErrorGroup());
System.out.println("systemTime=" + response.getSystemTime());
System.out.println("paymentId=" + response.getPaymentId());
System.out.println("price=" + response.getPrice());
System.out.println("paidPrice=" + response.getPaidPrice());
System.out.println("currency=" + response.getCurrency());
System.out.println("installment=" + response.getInstallment());
System.out.println("basketId=" + response.getBasketId());
System.out.println("binNumber=" + response.getBinNumber());
System.out.println("cardAssociation=" + response.getCardAssociation());
System.out.println("cardFamily=" + response.getCardFamily());
System.out.println("cardType=" + response.getCardType());
System.out.println("cardToken=" + response.getCardToken());
System.out.println("cardUserKey=" + response.getCardUserKey());
System.out.println("fraudStatus=" + response.getFraudStatus());
System.out.println("iyziCommissionFee=" + response.getIyziCommissionFee());
System.out.println("iyziCommissionRateAmount=" + response.getIyziCommissionRateAmount());
System.out.println("merchantCommissionRate=" + response.getMerchantCommissionRate());
System.out.println("merchantCommissionRateAmount=" +
response.getMerchantCommissionRateAmount());

List<EcomPaymentItemTransactionDto> itemTransactions = response.getItemTransactions();

if (itemTransactions != null && !itemTransactions.isEmpty()) {
    for (EcomPaymentItemTransactionDto itemTransaction : itemTransactions) {
        System.out.println("paymentTransactionId=" + itemTransaction.getPaymentTransactionId());
        System.out.println("itemId=" + itemTransaction.getItemId());
        System.out.println("paidPrice=" + itemTransaction.getPaidPrice());
        System.out.println("price=" + itemTransaction.getPrice());
        System.out.println("transactionStatus=" + itemTransaction.getTransactionStatus());
        System.out.println("blockageRate=" + itemTransaction.getBlockageRate());
        System.out.println("blockageRateAmountMerchant=" +
itemTransaction.getBlockageRateAmountMerchant());
        System.out.println("blockageRateAmountSubMerchant=" +
itemTransaction.getBlockageRateAmountSubMerchant());
        System.out.println("blockageResolvedDate=" + itemTransaction.getBlockageResolvedDate());
        System.out.println("iyziCommissionFee=" + itemTransaction.getIyziCommissionFee());
        System.out.println("iyziCommissionRateAmount=" +
itemTransaction.getIyziCommissionRateAmount());
        System.out.println("merchantCommissionRate=" +
itemTransaction.getMerchantCommissionRate());
    }
}

```

```

System.out.println("merchantCommissionRateAmount=" +
itemTransaction.getMerchantCommissionRateAmount());
System.out.println("merchantPayoutAmount=" + itemTransaction.getMerchantPayoutAmount());
System.out.println("subMerchantKey=" + itemTransaction.getSubMerchantKey());
System.out.println("subMerchantPayoutAmount=" +
itemTransaction.getSubMerchantPayoutAmount());
System.out.println("subMerchantPayoutRate=" + itemTransaction.getSubMerchantPayoutRate());
System.out.println("subMerchantPrice=" + itemTransaction.getSubMerchantPrice());
}
}

```

### Example of Auth3DS Response:

```

conversationId=123456789
locale=tr
status=success
errorCode=null
errorMessage=null
errorGroup=null
systemTime=1444142311338
paymentId=457932
price=100
paidPrice=110
currency=TRY
installment=3
basketId=B67832
binNumber=554960
cardAssociation=MASTER_CARD
cardFamily=Bonus
cardType=CREDIT_CARD
cardToken=null
cardUserKey=null
fraudStatus=1
iyziCommissionFee=0.25000000
iyziCommissionRateAmount=4.51000000
merchantCommissionRate=10.00000000
merchantCommissionRateAmount=10
paymentTransactionId=1307
itemId=BI101
paidPrice=33.00000000
price=30
transactionStatus=1
blockageRate=7.00000000

```



blockageRateAmountMerchant=2.31000000  
blockageRateAmountSubMerchant=0  
blockageResolvedDate=2015-10-16 17:38:31  
iyziCommissionFee=0.07500000  
iyziCommissionRateAmount=1.35300000  
merchantCommissionRate=10.00000000  
merchantCommissionRateAmount=3.00000000  
merchantPayoutAmount=2.26200000  
subMerchantKey=l4FCFyclof0paTP6687dOoch9Po=  
subMerchantPayoutAmount=27  
subMerchantPayoutRate=90.00000000  
subMerchantPrice=27  
paymentTransactionId=1308  
itemId=BI102  
paidPrice=55.00000000  
price=50  
transactionStatus=1  
blockageRate=7.00000000  
blockageRateAmountMerchant=3.85000000  
blockageRateAmountSubMerchant=0  
blockageResolvedDate=2015-10-16 17:38:31  
iyziCommissionFee=0.12500000  
iyziCommissionRateAmount=2.25500000  
merchantCommissionRate=10.00000000  
merchantCommissionRateAmount=5.00000000  
merchantPayoutAmount=6.77000000  
subMerchantKey=lcb6bA9DWEF5e8yGJI0ulEcW1kw=  
subMerchantPayoutAmount=42  
subMerchantPayoutRate=84.00000000  
subMerchantPrice=42  
paymentTransactionId=1309  
itemId=BI103  
paidPrice=22.00000000  
price=20  
transactionStatus=1  
blockageRate=7.00000000  
blockageRateAmountMerchant=1.54000000  
blockageRateAmountSubMerchant=0  
blockageResolvedDate=2015-10-16 17:38:31  
iyziCommissionFee=0.05000000  
iyziCommissionRateAmount=0.90200000  
merchantCommissionRate=10.00000000  
merchantCommissionRateAmount=2.00000000

```
merchantPayoutAmount=1.50800000
subMerchantKey=D9V/MqIRitUzA4dutL+nCBvnWfs=
subMerchantPayoutAmount=18
subMerchantPayoutRate=90.00000000
subMerchantPrice=18
```

#### JSON format of auth3DS request example :

```
{
  "locale" : "tr",
  "conversationId" : "123456789",
  "paymentId" : "457932",
  "conversationData" : "KPeA/vSq2nspTXa3mIHveg=="
}
```

#### JSON format of initialize3DS response example :

```
{
  "status" : "success",
  "locale" : "tr",
  "systemTime" : 1444142311338,
  "conversationId" : "123456789",
  "price" : 100,
  "paidPrice" : 110,
  "currency" : TRY,
  "installment" : 3,
  "paymentId" : "457932",
  "fraudStatus" : 1,
  "merchantCommissionRate" : 10.00000000,
  "merchantCommissionRateAmount" : 10,
  "iyziCommissionRateAmount" : 4.51000000,
  "iyziCommissionFee" : 0.25000000,
  "cardType" : "CREDIT_CARD",
  "cardAssociation" : "MASTER_CARD",
  "cardFamily" : "Bonus",
  "binNumber" : "554960",
  "basketId" : "B67832",
  "itemTransactions" : [ {
    "itemId" : "BI101",
    "paymentTransactionId" : "1307",
```

```

"transactionStatus" : 1,
"price" : 30,
"paidPrice" : 33.00000000,
"merchantCommissionRate" : 10.00000000,
"merchantCommissionRateAmount" : 3.00000000,
"iyziCommissionRateAmount" : 1.35300000,
"iyziCommissionFee" : 0.07500000,
"blockageRate" : 7.00000000,
"blockageRateAmountMerchant" : 2.31000000,
"blockageRateAmountSubMerchant" : 0,
"blockageResolvedDate" : "2015-10-16 17:38:31",
"subMerchantKey" : "l4FCFycl0f0paTP6687dOoch9Po=",
"subMerchantPrice" : 27,
"subMerchantPayoutRate" : 90.00000000,
"subMerchantPayoutAmount" : 27,
"merchantPayoutAmount" : 2.26200000
}, {
"itemId" : "BI102",
"paymentTransactionId" : "1308",
"transactionStatus" : 1,
"price" : 50,
"paidPrice" : 55.00000000,
"merchantCommissionRate" : 10.00000000,
"merchantCommissionRateAmount" : 5.00000000,
"iyziCommissionRateAmount" : 2.25500000,
"iyziCommissionFee" : 0.12500000,
"blockageRate" : 7.00000000,
"blockageRateAmountMerchant" : 3.85000000,
"blockageRateAmountSubMerchant" : 0,
"blockageResolvedDate" : "2015-10-16 17:38:31",
"subMerchantKey" : "lCb6bA9DWEF5e8yGJI0ulEcW1kw=",
"subMerchantPrice" : 42,
"subMerchantPayoutRate" : 84.00000000,
"subMerchantPayoutAmount" : 42,
"merchantPayoutAmount" : 6.77000000
}, {
"itemId" : "BI103",
"paymentTransactionId" : "1309",
"transactionStatus" : 1,
"price" : 20,
"paidPrice" : 22.00000000,
"merchantCommissionRate" : 10.00000000,
"merchantCommissionRateAmount" : 2.00000000,

```

```

"iyziCommissionRateAmount" : 0.90200000,
"iyziCommissionFee" : 0.05000000,
"blockageRate" : 7.00000000,
"blockageRateAmountMerchant" : 1.54000000,
"blockageRateAmountSubMerchant" : 0,
"blockageResolvedDate" : "2015-10-16 17:38:31",
"subMerchantKey" : "D9V/MqIRitUzA4dutL+nCBvnWfs=",
"subMerchantPrice" : 18,
"subMerchantPayoutRate" : 90.00000000,
"subMerchantPayoutAmount" : 18,
"merchantPayoutAmount" : 1.50800000
}]
}

```

### 3.3.1.4.2 3D Secure Payment Process for Subscription and Listing Product (Same process for marketplace merchants and non marketplace merchants)

Except of **initialize3DS** API service endpoint, **callback** and **auth3DS** process same with 3DS payment process for product group. (4.3.1.4.1 3D Secure Payment Process for Product Group, Marketplaces Merchants)

#### **initialize3DS – Listing or Subscription Payment Group**

- API Service (Endpoint) Name: **/payment/iyzipos/initialize3ds/ecom**
- Method: **HTTP POST**

#### **Example of initialize3DS Request:**

```

EcomPaymentServiceClient client = new EcomPaymentServiceClientImpl(configuration);

EcomPaymentThreeDSInitializeRequest request = new EcomPaymentThreeDSInitializeRequest();
request.setLocale(RequestLocaleType.TR.getValue());
request.setConversationId("123456789");
request.setCallbackUrl("https://www.merchant.com/callback");
request.setPrice(new BigDecimal("100"));
request.setPaidPrice(new BigDecimal("100"));
request.setCurrency("TRY");
request.setInstallment(1);
request.setBasketId("B67832");
request.setPaymentChannel(PaymentChannelRequestType.WEB.name());

```

```
request.setPaymentGroup(PaymentGroupRequestType.LISTING.name());
```

```
EcomPaymentCardDto paymentCard = new EcomPaymentCardDto();
paymentCard.setCardHolderName("Sabri Onur Tüzün");
paymentCard.setCardNumber("554960*****7015");
paymentCard.setExpireMonth("**");
paymentCard.setExpireYear("20**");
paymentCard.setCvc("***");
paymentCard.setRegisterCard(0);
request.setPaymentCard(paymentCard);
```

```
EcomPaymentBuyerDto buyer = new EcomPaymentBuyerDto();
buyer.setId("BY789");
buyer.setName("Sabri Onur");
buyer.setSurname("Tüzün");
buyer.setGsmNumber("+905350000000");
buyer.setEmail("onur.tuzun@iyzico.com");
buyer.setIdentityNumber("74300864791");
buyer.setLastLoginDate("2015-10-05 12:43:35");
buyer.setRegistrationDate("2013-04-21 15:12:09");
buyer.setRegistrationAddress("Nidakule Göztepe İş Merkezi Merdivenköy Mah. Bora Sok. No:1 Kat:19
Bağımsız 70/73 Göztepe Kadıköy");
buyer.setIp("85.34.78.112");
buyer.setCity("İstanbul");
buyer.setCountry("Türkiye");
buyer.setZipCode("34732");
request.setBuyer(buyer);
```

```
EcomPaymentShippingAddressDto shippingAddress = new EcomPaymentShippingAddressDto();
shippingAddress.setContactName("Hakan Erdoğan");
shippingAddress.setCity("İstanbul");
shippingAddress.setCountry("Türkiye");
shippingAddress.setAddress("19 Mayıs Mah. İnönü Cad. No:45 Kozyatağı");
shippingAddress.setZipCode("34742");
request.setShippingAddress(shippingAddress);
```

```
EcomPaymentBillingAddressDto billingAddress = new EcomPaymentBillingAddressDto();
billingAddress.setContactName("Hakan Erdoğan");
billingAddress.setCity("İstanbul");
billingAddress.setCountry("Türkiye");
billingAddress.setAddress("19 Mayıs Mah. İnönü Cad. No:45 Kozyatağı");
billingAddress.setZipCode("34742");
request.setBillingAddress(billingAddress);
```

```

List<EcomPaymentBasketItemDto> basketItems = new ArrayList<EcomPaymentBasketItemDto>();
EcomPaymentBasketItemDto basketItem1 = new EcomPaymentBasketItemDto();
basketItem1.setId("BI101");
basketItem1.setName("Dükkan listeleme ve katalog");
basketItem1.setCategory1("Listeleme");
basketItem1.setCategory2("Dükkan");
basketItem1.setItemType(BasketItemRequestType.PHYSICAL.name());
basketItem1.setPrice(new BigDecimal("30"));
basketItems.add(basketItem1);

EcomPaymentBasketItemDto basketItem2 = new EcomPaymentBasketItemDto();
basketItem2.setId("BI102");
basketItem2.setName("Listeleme hizmeti");
basketItem2.setCategory1("Listeleme");
basketItem2.setCategory2("Listeleme");
basketItem2.setItemType(BasketItemRequestType.VIRTUAL.name());
basketItem2.setPrice(new BigDecimal("50"));
basketItems.add(basketItem2);

EcomPaymentBasketItemDto basketItem3 = new EcomPaymentBasketItemDto();
basketItem3.setId("BI103");
basketItem3.setName("Servis hizmeti");
basketItem3.setCategory1("Listeleme");
basketItem3.setCategory2("Servis");
basketItem3.setItemType(BasketItemRequestType.VIRTUAL.name());
basketItem3.setPrice(new BigDecimal("20"));
basketItems.add(basketItem3);
request.setBasketItems(basketItems);

EcomPaymentThreeDSInitializeResponse response = client.initializeThreeDS(request);

System.out.println("conversationId=" + response.getConversationId());
System.out.println("locale=" + response.getLocale());
System.out.println("status=" + response.getStatus());
System.out.println("errorCode=" + response.getErrorCode());
System.out.println("errorMessage=" + response.getErrorMessage());
System.out.println("errorGroup=" + response.getErrorGroup());
System.out.println("systemTime=" + response.getSystemTime());
System.out.println("threeDSHtmlContent=" + response.getThreeDSHtmlContent());

```

### Example of initialize3DS Response:

```

conversationId=123456789
locale=tr
status=success
errorCode=null
errorMessage=null
errorGroup=null
systemTime=1445238853435
threeDSHtmlContent=<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html>
<head>
<title>MDpay default response template for web</title>
</head>
<body bgcolor="#02014E" OnLoad="OnLoadEvent();" >
<form name="downloadForm"
action="https://3dsecure.garanti.com.tr/acs/0036542387077064147970095830/merchant/control/master"
method="POST">
<input type="hidden"
name="PaReq"
value="eJxVUdtugzAM/RXU10nEAKgciO15WGdtKpqq017ZCFqkcalAUbZ1y8pvWxP8TI2fHxs3B+1
UsIOyU4rga+qadKDcvJsNmEsogEANwjczLfqJP Bb6SavSkFdcD0kN2h+aXIMy1ZgKk+L1VowD0LmI7ICLJReJ
SJKYQhRSGMki4FIWiiRDz+5rJBcAMqqK1s9CB4bhRvATn+JY9v WU0L6vnfHL66sCiQ2heQxwqazUWNanf
NMvO2G3TZ536/jan1olk/J4ly/9PXn5qWalbEVmKWtEh5QRoHGDvApjaYeRXLhMS3sDGK//XAogAtgXI0
U1IzPpGkTs6m/FJqValXKQcSBsXxHqM51VSpTYRzeY8xUI4VjZO2L5GFj+Wz3KluzMAA/ZIHnRwx4hzCgAY
85QMwi3841Ftn+uVkjSi9CFiAxLYh10OS651N9O/+v486qk0=">
<input type="hidden"
name="TermUrl" value="https://spos.isbank.com.tr/fim/est3Dgate?msgid=55814007">
<input type="hidden"
name="MD"
value="554960:FF5B0BEDB0E17AC8087EBAB69B4FB5EBE8CB4CA2601451549B96212890F104F4
:4434:##700656608619">
<!-- To support javascript unaware/disabled browsers -->
<noscript>
<center>Please click the submit button below.<br>
<input type="submit" name="submit" value="Submit"></center>
</noscript>
</form>

<SCRIPT LANGUAGE="Javascript" >
<!-- about:blank -->

```

```

<!--
function OnLoadEvent() {
    document.downloadForm.submit();
}
//-->
</SCRIPT>
</body>
</html>

```

#### JSON format of initialize3DS request example :

```

{
  "locale" : "tr",
  "conversationId" : "123456789",
  "price" : 100,
  "paidPrice" : 100,
  "currency" : TRY,
  "installment" : 1,
  "paymentChannel" : "WEB",
  "basketId" : "B67832",
  "paymentGroup" : "LISTING",
  "paymentCard" : {
    "cardHolderName" : "Sabri Onur Tüzün",
    "cardNumber" : "554960*****7015",
    "expireYear" : "20**",
    "expireMonth" : "**",
    "cvc" : "***",
    "registerCard" : 0
  },
  "buyer" : {
    "id" : "BY789",
    "name" : "Sabri Onur",
    "surname" : "Tüzün",
    "identityNumber" : "74300864791",
    "email" : "onur.tuzun@iyzico.com",
    "gsmNumber" : "+905350000000",
    "registrationDate" : "2013-04-21 15:12:09",
    "lastLoginDate" : "2015-10-05 12:43:35",
    "registrationAddress" : "Nidakule Göztepe İş Merkezi Merdivenköy Mah. Bora Sok. No:1 Kat:19
Bağımsız 70/73 Göztepe Kadıköy",
    "city" : "İstanbul",
    "country" : "Türkiye",

```



```

"zipCode" : "34732",
"ip" : "85.34.78.112"
},
"shippingAddress" : {
  "address" : "19 Mayıs Mah. İnönü Cad. No:45 Kozyatağı",
  "zipCode" : "34742",
  "contactName" : "Hakan Erdoğan",
  "city" : "İstanbul",
  "country" : "Türkiye"
},
"billingAddress" : {
  "address" : "19 Mayıs Mah. İnönü Cad. No:45 Kozyatağı",
  "zipCode" : "34742",
  "contactName" : "Hakan Erdoğan",
  "city" : "İstanbul",
  "country" : "Türkiye"
},
"basketItems" : [ {
  "id" : "BI101",
  "price" : 30,
  "name" : "Dükkan listeleme ve katalog",
  "category1" : "Listeleme",
  "category2" : "Dükkan",
  "itemType" : "PHYSICAL"
}, {
  "id" : "BI102",
  "price" : 50,
  "name" : "Listeleme hizmeti",
  "category1" : "Listeleme",
  "category2" : "Listeleme",
  "itemType" : "VIRTUAL"
}, {
  "id" : "BI103",
  "price" : 20,
  "name" : "Servis hizmeti",
  "category1" : "Listeleme",
  "category2" : "Servis",
  "itemType" : "VIRTUAL"
} ],
"callbackUrl" : "https://www.merchant.com/callback"
}

```

### JSON format of initialize3DS response example :

```
{
  "status" : "success",
  "locale" : "tr",
  "systemTime" : 1445238853435,
  "conversationId" : "123456789",
  "threeDSHtmlContent" : "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN\"
  \"http://www.w3.org/TR/html4/strict.dtd>\n\n<html>\n<head>\n  <title>MDpay default response
  template for web</title>\n</head>\n<body bgcolor=\"#02014E\" OnLoad=\"OnLoadEvent();\"
  >\n<form
  name=\"downloadForm\" \n  action=\"https://3dsecure.garanti.com.tr/acs/0036542389782599331550048898/merchant/control/master\" \n  method=\"POST\">\n  <input
  type=\"hidden\" \n    name=\"PaReq\" \n    value=\"eJxVUctugzAQ/BWUD8APMNjRxlLaHIqqIpTm
  0B6RsRkK8oiBBvr1tQJ2pN3Ztc7u7NwOBmtd+9aDUZLeNNdlx+1VxabFWOchG EsVhKy7V6fJXxr05VNLY
  mPfd+uEbpcLv71i6+aCpBLAXqMkA0u6myrsSwkT9hlxyxvy/KjyNMkHdFrMGW7Y9psALkKKPJeS4oJI5glD8
  drStcMA5p5yCs3gzszPz2CsY9tYqGgdUrbK7A5I/pLgbXU6FpNUoR25TsCPbZnrW2F3fAeQ6E7JT0r615Ajz
  WeX5yvqreGYRxELKQBfzGnTlggllxhHHlUuHN7LnL9S2sa4YTMAG4Acm3Qcki03NIG/+7/C34mqkE=\">\n
  <input type=\"hidden\" \n    name=\"TermUrl\"
  value=\"https://spos.isbank.com.tr/fim/est3Dgate?msgid=55814479\">\n  <input
  type=\"hidden\" \n    name=\"MD\" \n    value=\"554960:AB3CA68F65E473063C526C352BFAAB0
  EBD46B96FADC3293D13E1493F1E9EC00C:3568:##700656608619\">\n  <!-- To support javascript
  unaware/disabled browsers -->\n  <noscript>\n    <center>Please click the submit button
  below.<br>\n    <input type=\"submit\" name=\"submit\"
  value=\"Submit\"></center>\n  </noscript>\n</form>\n\n<SCRIPT LANGUAGE=\"Javascript\" >\n  <!--
  about:blank -->\n  <!--\n  function OnLoadEvent() {\n    document.downloadForm.submit();\n  }\n  //--
  >\n</SCRIPT>\n</body>\n</html>\u0000"
}
```

#### 3.3.1.4.3 3D Secure Payment Process for Product Group, Non-Marketplaces Merchants

It is almostly same process with 3D Secure Payment Process for SUBSCRIPTION and LISTING

(4.3.1.4.2). Differences,

- BasketItemType must set as Physical for each item.
- subMerchantKey and subMerchantPrice shouldnt send within request.
- Payment group has to set as PRODUCT.

```
request.setPaymentGroup(PaymentGroupRequestType.PRODUCT.name());
```

### 3.3.2 Cancel

Regarding to bank's regulations and rules, merchants are able to cancel any payment within the same day that payment received (*Before banks perform their own reconciliation*). If cancel request has applied, payment will not shown in credit card statement.

paymentId parameter is required for cancel. Status parameter will return in reponse and if status value is success, amount of transaction (*price*) will return in response too.

- API Service (Endpoint) Name: **/payment/iyzipos/cancel**
- Method: **HTTP POST**

#### Request Example:

```
EcomPaymentServiceClient client = new EcomPaymentServiceClientImpl(configuration);

PaymentCancelRequest request = new PaymentCancelRequest();
request.setLocale(RequestLocaleType.TR.getValue());
request.setConversationId("123456789");
request.setPaymentId("3031");
request.setIp("85.34.78.112");

PaymentCancelResponse response = client.cancel(request);

System.out.println("conversationId=" + response.getConversationId());
System.out.println("locale=" + response.getLocale());
System.out.println("status=" + response.getStatus());
System.out.println("errorCode=" + response.getErrorCode());
System.out.println("errorMessage=" + response.getErrorMessage());
System.out.println("errorGroup=" + response.getErrorGroup());
System.out.println("systemTime=" + response.getSystemTime());
System.out.println("paymentId=" + response.getPaymentId());
System.out.println("price=" + response.getPrice());
```

#### Response Example :

```
conversationId=123456789
locale=tr
status=success
```

```
errorCode=null
errorMessage=null
errorGroup=null
systemTime=1445239960166
paymentId=3031
price=1.10000000
```

#### JSON format of request example :

```
{
  "locale" : "tr",
  "conversationId" : "123456789",
  "paymentId" : "3031",
  "ip" : "85.34.78.112"
}
```

#### JSON format of response example :

```
{
  "status" : "success",
  "locale" : "tr",
  "systemTime" : 1445239960166,
  "conversationId" : "123456789",
  "paymentId" : "3031",
  "price" : 1.10000000
}
```

### 3.3.3 Refund

Merchants are able to refund any payment 24/7, 365 days.

Refund transactions will shown in buyer's card statement. Merchants are able to refund all amount or partial amount of any payment.

Refund amount can't be greater than total amount.

- item paidPrice =100 TL , refundPrice can not be greater than 100 TL.

- item paidPrice =100 TL, refundPrice 80 TL = price after refund will be 20 TL. For next refund request, refundPrice must be between 0.01 TL and 20 TL.

### 3.3.3.1 Standard Refund

Breakdown (*paymentTransactionId*) of a payment and its *price* are mandatory values to make refund. If status is success of a refund transaction, amount of *refundprice* will return in response. *refundprice* must less than or equal to refundable amount of payment and (*itemTransaction* → *paidPrice*)

- API Service (Endpoint) Name: **/payment/iyzipos/refund**
- Method: **HTTP POST**

#### Request Example:

```
EcomPaymentServiceClient client = new EcomPaymentServiceClientImpl(configuration);

PaymentRefundRequest request = new PaymentRefundRequest();
request.setLocale(RequestLocaleType.TR.getValue());
request.setConversationId("123456789");
request.setPaymentTransactionId("1543");
request.setPrice(new BigDecimal("3.3"));
request.setCurrency("TRY");
request.setIp("85.34.78.112");

PaymentRefundResponse response = client.refund(request);

System.out.println("conversationId=" + response.getConversationId());
System.out.println("locale=" + response.getLocale());
System.out.println("status=" + response.getStatus());
System.out.println("errorCode=" + response.getErrorCode());
System.out.println("errorMessage=" + response.getErrorMessage());
System.out.println("errorGroup=" + response.getErrorGroup());
System.out.println("systemTime=" + response.getSystemTime());
System.out.println("paymentId=" + response.getPaymentId());
System.out.println("paymentTransactionId=" + response.getPaymentTransactionId());
```

```
System.out.println("price=" + response.getPrice());  
System.out.println("currency=" + response.getCurrency());
```

#### Response Example :

```
conversationId=123456789  
locale=tr  
status=success  
errorCode=null  
errorMessage=null  
errorGroup=null  
systemTime=1445240504793  
paymentId=3032  
paymentTransactionId=1543  
price=3.3  
currency=TRY
```

#### JSON format of request example :

```
{  
  "locale" : "tr",  
  "conversationId" : "123456789",  
  "paymentTransactionId" : "1543",  
  "price" : 3.3,  
  "currency" : TRY,  
  "ip" : "85.34.78.112"  
}
```

#### JSON format of response example :

```
{  
  "status" : "success",  
  "locale" : "tr",  
  "systemTime" : 1445240504793,  
  "conversationId" : "123456789",  
  "paymentId" : "3032",  
  "paymentTransactionId" : "1543",  
  "price" : 3.3
```

### 3.3.3.2 Refund after settlement

If refund request for a breakdown of a payment receive after settlement, Merchants must collect all payments. (Payout already done to submerchants.)

E.g. : price is 100 TL and settlement done. Breakdown of payment listed below. (marketplace Merchant)

- 90 TL for submerchant
- 8 TL for Merchant
- 2 TL for iyzico

Lets consider Refund request as 50 TL. In this case amount needed to refund as listed below,

- 45 TL from submerchant,
- 4 TL from merchant,
- 1 TL from iyzico.

In that case iyzico manage refund after settlement process. Manage netting operation during the next settlement process to merchant / submerchant.

- If submerchant balance is enough to cover it, refund will accept and record as negative side of submerchant balance. (have amount that waiting for payout or waiting for approval)
- If merchant balance is enough to cover it, refund will accept and record as negative side of merchant balance.

If submerchant balance isn't enough to cover refund, iyzico returns an error message to merchant as "**5116 error code**". In that case merchants able to accept refund behalf of submerchant, by using service endpoint shown below.

If merchant balance doesnt enough to cover refund, iyzico returns error message to merchant under **5117 error code**. In that case refund can not accept.

- **API Service (Endpoint) Name: /payment/iyzipos/refund/merchant/charge**

- Method: **HTTP POST**

### 3.3.4 Pre Authorization

Pre Auth transactions are completely same with auth transactions in terms of request parameters but API endpoint service is different. Unlike auth transaction amount is not captured from cardholder account in Pre Auth transactions.

- API Service (Endpoint) Name: **/payment/iyzipos/preauth**
- Method: **HTTP POST**

#### Example Java Request:

```
EcomPaymentServiceClient client = new EcomPaymentServiceClientImpl(configuration);

EcomPaymentPreAuthRequest request = new EcomPaymentPreAuthRequest();
request.setLocale(RequestLocaleType.TR.getValue());
request.setConversationId("123456789");
request.setPrice(new BigDecimal("100"));
request.setPaidPrice(new BigDecimal("110"));
request.setCurrency("TRY");
request.setInstallment(3);
request.setBasketId("B67832");
request.setPaymentChannel(PaymentChannelRequestType.WEB.name());
request.setPaymentGroup(PaymentGroupRequestType.PRODUCT.name());

EcomPaymentCardDto paymentCard = new EcomPaymentCardDto();
paymentCard.setCardHolderName("Sabri Onur Tüzün");
paymentCard.setCardNumber("554960*****");
paymentCard.setExpireMonth("**");
paymentCard.setExpireYear("20**");
paymentCard.setCvc("***");
paymentCard.setRegisterCard(0);
request.setPaymentCard(paymentCard);

EcomPaymentBuyerDto buyer = new EcomPaymentBuyerDto();
buyer.setId("BY789");
buyer.setName("Sabri Onur");
buyer.setSurname("Tüzün");
buyer.setGsmNumber("+905350000000");
buyer.setEmail("email@buyeremail.com");
```



```

buyer.setIdentityNumber("74300864791");
buyer.setLastLoginDate("2015-10-05 12:43:35");
buyer.setRegistrationDate("2013-04-21 15:12:09");
buyer.setRegistrationAddress("Nidakule Göztepe İş Merkezi Merdivenköy Mah. Bora Sok. No:1
Kat:19 Bağımsız 70/73 Göztepe Kadıköy");
buyer.setIp("85.34.78.112");
buyer.setCity("İstanbul");
buyer.setCountry("Türkiye");
buyer.setZipCode("34732");
request.setBuyer(buyer);

```

```

EcomPaymentShippingAddressDto shippingAddress = new EcomPaymentShippingAddressDto();
shippingAddress.setContactName("Hakan Erdoğan");
shippingAddress.setCity("İstanbul");
shippingAddress.setCountry("Türkiye");
shippingAddress.setAddress("19 Mayıs Mah. İnönü Cad. No:45 Kozyatağı");
shippingAddress.setZipCode("34742");
request.setShippingAddress(shippingAddress);

```

```

EcomPaymentBillingAddressDto billingAddress = new EcomPaymentBillingAddressDto();
billingAddress.setContactName("Hakan Erdoğan");
billingAddress.setCity("İstanbul");
billingAddress.setCountry("Türkiye");
billingAddress.setAddress("19 Mayıs Mah. İnönü Cad. No:45 Kozyatağı");
billingAddress.setZipCode("34742");
request.setBillingAddress(billingAddress);

```

```

List<EcomPaymentBasketItemDto> basketItems = new ArrayList<EcomPaymentBasketItemDto>();
EcomPaymentBasketItemDto basketItem1 = new EcomPaymentBasketItemDto();
basketItem1.setId("BI101");
basketItem1.setName("ABC Marka Kolye");
basketItem1.setCategory1("Giyim");
basketItem1.setCategory2("Aksesuar");
basketItem1.setItemType(BasketItemRequestType.PHYSICAL.name());
basketItem1.setPrice(new BigDecimal("30"));
basketItem1.setSubMerchantKey("I4FCFyclof0paTP6687dOoch9Po=");
basketItem1.setSubMerchantPrice(new BigDecimal("27"));
basketItems.add(basketItem1);

```

```

EcomPaymentBasketItemDto basketItem2 = new EcomPaymentBasketItemDto();
basketItem2.setId("BI102");
basketItem2.setName("XYZ Oyun Kodu");
basketItem2.setCategory1("Oyun");
basketItem2.setCategory2("Online Oyun Kodları");

```

```

basketItem2.setItemType(BasketItemRequestType.VIRTUAL.name());
basketItem2.setPrice(new BigDecimal("50"));
basketItem2.setSubMerchantKey("lcB6bA9DWEF5e8yGJI0uIEcW1kw=");
basketItem2.setSubMerchantPrice(new BigDecimal("42"));
basketItems.add(basketItem2);

EcomPaymentBasketItemDto basketItem3 = new EcomPaymentBasketItemDto();
basketItem3.setId("BI103");
basketItem3.setName("EDC Marka Usb");
basketItem3.setCategory1("Elektronik");
basketItem3.setCategory2("Usb / Cable");
basketItem3.setItemType(BasketItemRequestType.PHYSICAL.name());
basketItem3.setPrice(new BigDecimal("20"));
basketItem3.setSubMerchantKey("D9V/MqIRitUzA4dutL+nCBvnWfs=");
basketItem3.setSubMerchantPrice(new BigDecimal("18"));
basketItems.add(basketItem3);
request.setBasketItems(basketItems);

EcomPaymentPreAuthResponse response = client.preAuth(request);

System.out.println("conversationId=" + response.getConversationId());
System.out.println("locale=" + response.getLocale());
System.out.println("status=" + response.getStatus());
System.out.println("errorCode=" + response.getErrorCode());
System.out.println("errorMessage=" + response.getErrorMessage());
System.out.println("errorGroup=" + response.getErrorGroup());
System.out.println("systemTime=" + response.getSystemTime());
System.out.println("paymentId=" + response.getPaymentId());
System.out.println("price=" + response.getPrice());
System.out.println("paidPrice=" + response.getPaidPrice());
System.out.println("currency=" + response.getCurrency());
System.out.println("installment=" + response.getInstallment());
System.out.println("basketId=" + response.getBasketId());
System.out.println("binNumber=" + response.getBinNumber());
System.out.println("cardAssociation=" + response.getCardAssociation());
System.out.println("cardFamily=" + response.getCardFamily());
System.out.println("cardType=" + response.getCardType());
System.out.println("cardToken=" + response.getCardToken());
System.out.println("cardUserKey=" + response.getCardUserKey());
System.out.println("fraudStatus=" + response.getFraudStatus());
System.out.println("iyziCommissionFee=" + response.getIyziCommissionFee());
System.out.println("iyziCommissionRateAmount=" + response.getIyziCommissionRateAmount());
System.out.println("merchantCommissionRate=" + response.getMerchantCommissionRate());
System.out.println("merchantCommissionRateAmount=" +

```

```

response.getMerchantCommissionRateAmount());

List<EcomPaymentItemTransactionDto> itemTransactions = response.getItemTransactions();

if (itemTransactions != null && !itemTransactions.isEmpty()) {
    for (EcomPaymentItemTransactionDto itemTransaction : itemTransactions) {
        System.out.println("paymentTransactionId=" + itemTransaction.getPaymentTransactionId());
        System.out.println("itemId=" + itemTransaction.getItemId());
        System.out.println("paidPrice=" + itemTransaction.getPaidPrice());
        System.out.println("price=" + itemTransaction.getPrice());
        System.out.println("transactionStatus=" + itemTransaction.getTransactionStatus());
        System.out.println("blockageRate=" + itemTransaction.getBlockageRate());
        System.out.println("blockageRateAmountMerchant=" +
itemTransaction.getBlockageRateAmountMerchant());
        System.out.println("blockageRateAmountSubMerchant=" +
itemTransaction.getBlockageRateAmountSubMerchant());
        System.out.println("blockageResolvedDate=" + itemTransaction.getBlockageResolvedDate());
        System.out.println("iyziCommissionFee=" + itemTransaction.getIyziCommissionFee());
        System.out.println("iyziCommissionRateAmount=" +
itemTransaction.getIyziCommissionRateAmount());
        System.out.println("merchantCommissionRate=" +
itemTransaction.getMerchantCommissionRate());
        System.out.println("merchantCommissionRateAmount=" +
itemTransaction.getMerchantCommissionRateAmount());
        System.out.println("merchantPayoutAmount=" +
itemTransaction.getMerchantPayoutAmount());
        System.out.println("subMerchantKey=" + itemTransaction.getSubMerchantKey());
        System.out.println("subMerchantPayoutAmount=" +
itemTransaction.getSubMerchantPayoutAmount());
        System.out.println("subMerchantPayoutRate=" +
itemTransaction.getSubMerchantPayoutRate());
        System.out.println("subMerchantPrice=" + itemTransaction.getSubMerchantPrice());
    }
}

```

### Example Java Response:

```

conversationId=123456789
locale=tr
status=success
errorCode=null
errorMessage=null

```

```

errorGroup=null
systemTime=1444142311338
paymentId=2744
price=100
paidPrice=110
currency=TRY
installment=3
basketId=B67832
binNumber=554960
cardAssociation=MASTER_CARD
cardFamily=Bonus
cardType=CREDIT_CARD
cardToken=null
cardUserKey=null
fraudStatus=1
iyziCommissionFee=0.25000000
iyziCommissionRateAmount=4.51000000
merchantCommissionRate=10.00000000
merchantCommissionRateAmount=10
paymentTransactionId=1307
itemId=BI101
paidPrice=33.00000000
price=30
transactionStatus=1
blockageRate=7.00000000
blockageRateAmountMerchant=2.31000000
blockageRateAmountSubMerchant=0
blockageResolvedDate=2015-10-16 17:38:31
iyziCommissionFee=0.07500000
iyziCommissionRateAmount=1.35300000
merchantCommissionRate=10.00000000
merchantCommissionRateAmount=3.00000000
merchantPayoutAmount=2.26200000
subMerchantKey=l4FCFyclof0paTP6687dOoch9Po=
subMerchantPayoutAmount=27
subMerchantPayoutRate=90.00000000
subMerchantPrice=27
paymentTransactionId=1308
itemId=BI102
paidPrice=55.00000000
price=50
transactionStatus=1
blockageRate=7.00000000
blockageRateAmountMerchant=3.85000000

```

```

blockageRateAmountSubMerchant=0
blockageResolvedDate=2015-10-16 17:38:31
iyziCommissionFee=0.12500000
iyziCommissionRateAmount=2.25500000
merchantCommissionRate=10.00000000
merchantCommissionRateAmount=5.00000000
merchantPayoutAmount=6.77000000
subMerchantKey=lcB6bA9DWEF5e8yGJI0uIEcW1kw=
subMerchantPayoutAmount=42
subMerchantPayoutRate=84.00000000
subMerchantPrice=42
paymentTransactionId=1309
itemId=BI103
paidPrice=22.00000000
price=20
transactionStatus=1
blockageRate=7.00000000
blockageRateAmountMerchant=1.54000000
blockageRateAmountSubMerchant=0
blockageResolvedDate=2015-10-16 17:38:31
iyziCommissionFee=0.05000000
iyziCommissionRateAmount=0.90200000
merchantCommissionRate=10.00000000
merchantCommissionRateAmount=2.00000000
merchantPayoutAmount=1.50800000
subMerchantKey=D9V/MqIRitUzA4dutL+nCBvnWfs=
subMerchantPayoutAmount=18
subMerchantPayoutRate=90.00000000
subMerchantPrice=18

```

#### Example Request (JSON) :

```

{
  "locale" : "tr",
  "conversationId" : "123456789",
  "price" : 100,
  "paidPrice" : 110,
  "currency" : TRY,
  "installment" : 3,
  "paymentChannel" : "WEB",
  "basketId" : "B67832",
  "paymentGroup" : "PRODUCT",
  "paymentCard" : {

```

```
"cardHolderName" : "Sabri Onur Tüzün",
"cardNumber" : "554960*****",
"expireYear" : "20**",
"expireMonth" : "**",
"cvc" : "***",
"registerCard" : 0
},
"buyer" : {
  "id" : "BY789",
  "name" : "Sabri Onur",
  "surname" : "Tüzün",
  "identityNumber" : "74300864791",
  "email" : "email@buyeremail.com",
  "gsmNumber" : "+905350000000",
  "registrationDate" : "2013-04-21 15:12:09",
  "lastLoginDate" : "2015-10-05 12:43:35",
  "registrationAddress" : "Nidakule Göztepe İş Merkezi Merdivenköy Mah. Bora Sok. No:1 Kat:19
Bağımsız 70/73 Göztepe Kadıköy",
  "city" : "İstanbul",
  "country" : "Türkiye",
  "zipCode" : "34732",
  "ip" : "85.34.78.112"
},
"shippingAddress" : {
  "address" : "19 Mayıs Mah. İnönü Cad. No:45 Kozyatağı",
  "zipCode" : "34742",
  "contactName" : "Hakan Erdoğan",
  "city" : "İstanbul",
  "country" : "Türkiye"
},
"billingAddress" : {
  "address" : "19 Mayıs Mah. İnönü Cad. No:45 Kozyatağı",
  "zipCode" : "34742",
  "contactName" : "Hakan Erdoğan",
  "city" : "İstanbul",
  "country" : "Türkiye"
},
"basketItems" : [ {
  "id" : "BI101",
  "price" : 30,
  "name" : "ABC Marka Kolye",
  "category1" : "Giyim",
  "category2" : "Aksesuar",
  "itemType" : "PHYSICAL",
```

```

"subMerchantKey" : "l4FCFyclof0paTP6687dOoch9Po=",
"subMerchantPrice" : 27
}, {
  "id" : "BI102",
  "price" : 50,
  "name" : "XYZ Oyun Kodu",
  "category1" : "Oyun",
  "category2" : "Online Oyun Kodları",
  "itemType" : "VIRTUAL",
  "subMerchantKey" : "lCb6bA9DWEF5e8yGJI0ulEcW1kw=",
  "subMerchantPrice" : 42
}, {
  "id" : "BI103",
  "price" : 20,
  "name" : "EDC Marka Usb",
  "category1" : "Elektronik",
  "category2" : "Usb / Cable",
  "itemType" : "PHYSICAL",
  "subMerchantKey" : "D9V/MqIRitUzA4dutL+nCBvnWfs=",
  "subMerchantPrice" : 18
}]
}

```

#### Example Response (JSON):

```

{
  "status" : "success",
  "locale" : "tr",
  "systemTime" : 1444142311338,
  "conversationId" : "123456789",
  "price" : 100,
  "paidPrice" : 110,
  "installment" : 3,
  "paymentId" : "2744",
  "fraudStatus" : 1,
  "merchantCommissionRate" : 10.00000000,
  "merchantCommissionRateAmount" : 10,
  "iyziCommissionRateAmount" : 4.51000000,
  "iyziCommissionFee" : 0.25000000,
  "cardType" : "CREDIT_CARD",
  "cardAssociation" : "MASTER_CARD",
  "cardFamily" : "Bonus",
  "binNumber" : "554960",

```

```

"basketId" : "B67832",
"itemTransactions" : [ {
  "itemId" : "BI101",
  "paymentTransactionId" : "1307",
  "transactionStatus" : 1,
  "price" : 30,
  "paidPrice" : 33.00000000,
  "merchantCommissionRate" : 10.00000000,
  "merchantCommissionRateAmount" : 3.00000000,
  "iyziCommissionRateAmount" : 1.35300000,
  "iyziCommissionFee" : 0.07500000,
  "blockageRate" : 7.00000000,
  "blockageRateAmountMerchant" : 2.31000000,
  "blockageRateAmountSubMerchant" : 0,
  "blockageResolvedDate" : "2015-10-16 17:38:31",
  "subMerchantKey" : "l4FCFyclof0paTP6687dOoch9Po=",
  "subMerchantPrice" : 27,
  "subMerchantPayoutRate" : 90.00000000,
  "subMerchantPayoutAmount" : 27,
  "merchantPayoutAmount" : 2.26200000
}, {
  "itemId" : "BI102",
  "paymentTransactionId" : "1308",
  "transactionStatus" : 1,
  "price" : 50,
  "paidPrice" : 55.00000000,
  "merchantCommissionRate" : 10.00000000,
  "merchantCommissionRateAmount" : 5.00000000,
  "iyziCommissionRateAmount" : 2.25500000,
  "iyziCommissionFee" : 0.12500000,
  "blockageRate" : 7.00000000,
  "blockageRateAmountMerchant" : 3.85000000,
  "blockageRateAmountSubMerchant" : 0,
  "blockageResolvedDate" : "2015-10-16 17:38:31",
  "subMerchantKey" : "lcb6bA9DWEF5e8yGJI0ulEcW1kw=",
  "subMerchantPrice" : 42,
  "subMerchantPayoutRate" : 84.00000000,
  "subMerchantPayoutAmount" : 42,
  "merchantPayoutAmount" : 6.77000000
}, {
  "itemId" : "BI103",
  "paymentTransactionId" : "1309",
  "transactionStatus" : 1,
  "price" : 20,

```



```

"paidPrice" : 22.00000000,
"merchantCommissionRate" : 10.00000000,
"merchantCommissionRateAmount" : 2.00000000,
"iyziCommissionRateAmount" : 0.90200000,
"iyziCommissionFee" : 0.05000000,
"blockageRate" : 7.00000000,
"blockageRateAmountMerchant" : 1.54000000,
"blockageRateAmountSubMerchant" : 0,
"blockageResolvedDate" : "2015-10-16 17:38:31",
"subMerchantKey" : "D9V/MqIRitUzA4dutL+nCBvnWfs=",
"subMerchantPrice" : 18,
"subMerchantPayoutRate" : 90.00000000,
"subMerchantPayoutAmount" : 18,
"merchantPayoutAmount" : 1.50800000
}]
}

```

### 3.3.5 Post Authorization

Post Auth requests have to send to after Pre Auth transactions to capture the amount. If amount parameter is not specify in request, total amount will captured which is send within Pre Auth request. It is possible to make capture less than the amount which is send within Pre Auth request.

- API Service (Endpoint) Name: **/payment/iyzipos/postauth**
- Method: **HTTP POST**

#### Example Request :

```

EcomPaymentServiceClient client = new EcomPaymentServiceClientImpl(configuration);

PaymentPostAuthRequest request = new PaymentPostAuthRequest();
request.setLocale(RequestLocaleType.TR.getValue());
request.setConversationId("123456789");
request.setPaymentId("457932");
request.setIp("85.34.78.112");
request.setPaidPrice(new BigDecimal("110"));

```

```

PaymentPostAuthResponse response = client.postAuth(request);

System.out.println("conversationId=" + response.getConversationId());
System.out.println("locale=" + response.getLocale());
System.out.println("status=" + response.getStatus());
System.out.println("errorCode=" + response.getErrorCode());
System.out.println("errorMessage=" + response.getErrorMessage());
System.out.println("errorGroup=" + response.getErrorGroup());
System.out.println("systemTime=" + response.getSystemTime());
System.out.println("paymentId=" + response.getPaymentId());
System.out.println("price=" + response.getPrice());
System.out.println("paidPrice=" + response.getPaidPrice());
System.out.println("currency=" + response.getCurrency());
System.out.println("installment=" + response.getInstallment());
System.out.println("basketId=" + response.getBasketId());
System.out.println("binNumber=" + response.getBinNumber());
System.out.println("cardAssociation=" + response.getCardAssociation());
System.out.println("cardFamily=" + response.getCardFamily());
System.out.println("cardType=" + response.getCardType());
System.out.println("cardToken=" + response.getCardToken());
System.out.println("cardUserKey=" + response.getCardUserKey());
System.out.println("fraudStatus=" + response.getFraudStatus());
System.out.println("iyziCommissionFee=" + response.getIyziCommissionFee());
System.out.println("iyziCommissionRateAmount=" + response.getIyziCommissionRateAmount());
System.out.println("merchantCommissionRate=" + response.getMerchantCommissionRate());
System.out.println("merchantCommissionRateAmount=" +
response.getMerchantCommissionRateAmount());

List<EcomPaymentItemTransactionDto> itemTransactions = response.getItemTransactions();

if (itemTransactions != null && !itemTransactions.isEmpty()) {
    for (EcomPaymentItemTransactionDto itemTransaction : itemTransactions) {
        System.out.println("paymentTransactionId=" + itemTransaction.getPaymentTransactionId());
        System.out.println("itemId=" + itemTransaction.getItemId());
        System.out.println("paidPrice=" + itemTransaction.getPaidPrice());
        System.out.println("price=" + itemTransaction.getPrice());
        System.out.println("transactionStatus=" + itemTransaction.getTransactionStatus());
        System.out.println("blockageRate=" + itemTransaction.getBlockageRate());
        System.out.println("blockageRateAmountMerchant=" +
itemTransaction.getBlockageRateAmountMerchant());
        System.out.println("blockageRateAmountSubMerchant=" +
itemTransaction.getBlockageRateAmountSubMerchant());
    }
}

```

```

System.out.println("blockageResolvedDate=" + itemTransaction.getBlockageResolvedDate());
System.out.println("iyziCommissionFee=" + itemTransaction.getIyziCommissionFee());
System.out.println("iyziCommissionRateAmount=" +
itemTransaction.getIyziCommissionRateAmount());
System.out.println("merchantCommissionRate=" +
itemTransaction.getMerchantCommissionRate());
System.out.println("merchantCommissionRateAmount=" +
itemTransaction.getMerchantCommissionRateAmount());
System.out.println("merchantPayoutAmount=" + itemTransaction.getMerchantPayoutAmount());
System.out.println("subMerchantKey=" + itemTransaction.getSubMerchantKey());
System.out.println("subMerchantPayoutAmount=" +
itemTransaction.getSubMerchantPayoutAmount());
System.out.println("subMerchantPayoutRate=" + itemTransaction.getSubMerchantPayoutRate());
System.out.println("subMerchantPrice=" + itemTransaction.getSubMerchantPrice());
}
}

```

#### Example Response:

```

conversationId=123456789
locale=tr
status=success
errorCode=null
errorMessage=null
errorGroup=null
systemTime=1444142311338
paymentId=457932
price=100
paidPrice=110
currency=TRY
installment=3
basketId=B67832
binNumber=554960
cardAssociation=MASTER_CARD
cardFamily=Bonus
cardType=CREDIT_CARD
cardToken=null
cardUserKey=null
fraudStatus=1
iyziCommissionFee=0.25000000
iyziCommissionRateAmount=4.51000000
merchantCommissionRate=10.00000000

```

```

merchantCommissionRateAmount=10
paymentTransactionId=1307
itemId=BI101
paidPrice=33.00000000
price=30
transactionStatus=1
blockageRate=7.00000000
blockageRateAmountMerchant=2.31000000
blockageRateAmountSubMerchant=0
blockageResolvedDate=2015-10-16 17:38:31
iyziCommissionFee=0.07500000
iyziCommissionRateAmount=1.35300000
merchantCommissionRate=10.00000000
merchantCommissionRateAmount=3.00000000
merchantPayoutAmount=2.26200000
subMerchantKey=l4FCFyclof0paTP6687dOoch9Po=
subMerchantPayoutAmount=27
subMerchantPayoutRate=90.00000000
subMerchantPrice=27
paymentTransactionId=1308
itemId=BI102
paidPrice=55.00000000
price=50
transactionStatus=1
blockageRate=7.00000000
blockageRateAmountMerchant=3.85000000
blockageRateAmountSubMerchant=0
blockageResolvedDate=2015-10-16 17:38:31
iyziCommissionFee=0.12500000
iyziCommissionRateAmount=2.25500000
merchantCommissionRate=10.00000000
merchantCommissionRateAmount=5.00000000
merchantPayoutAmount=6.77000000
subMerchantKey=lcB6bA9DWEF5e8yGJI0uIEcW1kw=
subMerchantPayoutAmount=42
subMerchantPayoutRate=84.00000000
subMerchantPrice=42
paymentTransactionId=1309
itemId=BI103
paidPrice=22.00000000
price=20
transactionStatus=1
blockageRate=7.00000000
blockageRateAmountMerchant=1.54000000

```

```

blockageRateAmountSubMerchant=0
blockageResolvedDate=2015-10-16 17:38:31
iyziCommissionFee=0.05000000
iyziCommissionRateAmount=0.90200000
merchantCommissionRate=10.00000000
merchantCommissionRateAmount=2.00000000
merchantPayoutAmount=1.50800000
subMerchantKey=D9V/MqIRitUzA4dutL+nCBvnWfs=
subMerchantPayoutAmount=18
subMerchantPayoutRate=90.00000000
subMerchantPrice=18

```

#### Example Request (JSON):

```

{
  "locale" : "tr",
  "conversationId" : "123456789",
  "paymentId" : "457932",
  "paidPrice" : 110,
  "currency" : TRY,
  "ip" : "85.34.78.112"
}

```

#### Example Response (JSON):

```

{
  "status" : "success",
  "locale" : "tr",
  "systemTime" : 1444142311338,
  "conversationId" : "123456789",
  "price" : 100,
  "paidPrice" : 110,
  "currency" : TRY,
  "installment" : 3,
  "paymentId" : "457932",
  "fraudStatus" : 1,
  "merchantCommissionRate" : 10.00000000,
  "merchantCommissionRateAmount" : 10,
  "iyziCommissionRateAmount" : 4.51000000,

```

```

"iyziCommissionFee" : 0.25000000,
"cardType" : "CREDIT_CARD",
"cardAssociation" : "MASTER_CARD",
"cardFamily" : "Bonus",
"binNumber" : "554960",
"basketId" : "B67832",
"itemTransactions" : [ {
  "itemId" : "BI101",
  "paymentTransactionId" : "1307",
  "transactionStatus" : 1,
  "price" : 30,
  "paidPrice" : 33.00000000,
  "merchantCommissionRate" : 10.00000000,
  "merchantCommissionRateAmount" : 3.00000000,
  "iyziCommissionRateAmount" : 1.35300000,
  "iyziCommissionFee" : 0.07500000,
  "blockageRate" : 7.00000000,
  "blockageRateAmountMerchant" : 2.31000000,
  "blockageRateAmountSubMerchant" : 0,
  "blockageResolvedDate" : "2015-10-16 17:38:31",
  "subMerchantKey" : "l4FCFyclof0paTP6687dOoch9Po=",
  "subMerchantPrice" : 27,
  "subMerchantPayoutRate" : 90.00000000,
  "subMerchantPayoutAmount" : 27,
  "merchantPayoutAmount" : 2.26200000
}, {
  "itemId" : "BI102",
  "paymentTransactionId" : "1308",
  "transactionStatus" : 1,
  "price" : 50,
  "paidPrice" : 55.00000000,
  "merchantCommissionRate" : 10.00000000,
  "merchantCommissionRateAmount" : 5.00000000,
  "iyziCommissionRateAmount" : 2.25500000,
  "iyziCommissionFee" : 0.12500000,
  "blockageRate" : 7.00000000,
  "blockageRateAmountMerchant" : 3.85000000,
  "blockageRateAmountSubMerchant" : 0,
  "blockageResolvedDate" : "2015-10-16 17:38:31",
  "subMerchantKey" : "lcB6bA9DWEF5e8yGJI0uIEcW1kw=",
  "subMerchantPrice" : 42,
  "subMerchantPayoutRate" : 84.00000000,
  "subMerchantPayoutAmount" : 42,

```

```

"merchantPayoutAmount" : 6.77000000
}, {
  "itemId" : "BI103",
  "paymentTransactionId" : "1309",
  "transactionStatus" : 1,
  "price" : 20,
  "paidPrice" : 22.00000000,
  "merchantCommissionRate" : 10.00000000,
  "merchantCommissionRateAmount" : 2.00000000,
  "iyziCommissionRateAmount" : 0.90200000,
  "iyziCommissionFee" : 0.05000000,
  "blockageRate" : 7.00000000,
  "blockageRateAmountMerchant" : 1.54000000,
  "blockageRateAmountSubMerchant" : 0,
  "blockageResolvedDate" : "2015-10-16 17:38:31",
  "subMerchantKey" : "D9V/MqIRitUzA4dutL+nCBvnWfs=",
  "subMerchantPrice" : 18,
  "subMerchantPayoutRate" : 90.00000000,
  "subMerchantPayoutAmount" : 18,
  "merchantPayoutAmount" : 1.50800000
}]
}

```

### 3.4 Approval for Payout (payment transfer)

For the marketplace solution, all payments keep in a secure account, until merchant send an approval for all breakdowns of a payment. In that period merchants are able to cancel the payment, refund partial amount or all amount of a breakdown, send approval if item was arrived to buyer without any problem, and also able to take back given approval.

#### 3.4.1 Item Approval

*paymentTransactionId* is a mandatory parameter for item approval.

- **API Service (Endpoint) Name:** `/payment/iyzipos/item/approve`
- **Method:** **HTTP POST**

#### Request Example :

```
EcomPaymentServiceClient client = new EcomPaymentServiceClientImpl(configuration);

ApprovalRequest request = new ApprovalRequest();
request.setLocale(RequestLocaleType.TR.getValue());
request.setConversationId("123456789");
request.setPaymentTransactionId("1546");

ApprovalResponse response = client.approve(request);

System.out.println("conversationId=" + response.getConversationId());
System.out.println("locale=" + response.getLocale());
System.out.println("status=" + response.getStatus());
System.out.println("errorCode=" + response.getErrorCode());
System.out.println("errorMessage=" + response.getErrorMessage());
System.out.println("errorGroup=" + response.getErrorGroup());
System.out.println("systemTime=" + response.getSystemTime());
System.out.println("paymentTransactionId=" + response.getPaymentTransactionId());
```

#### Response Example :

```
conversationId=123456789
locale=tr
status=success
errorCode=null
errorMessage=null
errorGroup=null
systemTime=1445240902151
paymentTransactionId=1546
```

#### JSON format of request example :

```
{
  "locale" : "tr",
  "conversationId" : "123456789",
  "paymentTransactionId" : "1546"
}
```

#### JSON format of response example :



```
{
  "status" : "success",
  "locale" : "tr",
  "systemTime" : 1445240902151,
  "conversationId" : "123456789",
  "paymentTransactionId" : "1546"
}
```

### 3.4.2 Rollback item approval

*paymentTransactionId* is a mandatory parameter for this transaction.

- API Service (Endpoint) Name: **/payment/iyzipos/item/disapprove**
- Method: **HTTP POST**

#### Request Example :

```
EcomPaymentServiceClient client = new EcomPaymentServiceClientImpl(configuration);

ApprovalRequest request = new ApprovalRequest();
request.setLocale(RequestLocaleType.TR.getValue());
request.setConversationId("123456789");
request.setPaymentTransactionId("1546");

ApprovalResponse response = client.disapprove(request);

System.out.println("conversationId=" + response.getConversationId());
System.out.println("locale=" + response.getLocale());
System.out.println("status=" + response.getStatus());
System.out.println("errorCode=" + response.getErrorCode());
System.out.println("errorMessage=" + response.getErrorMessage());
System.out.println("errorGroup=" + response.getErrorGroup());
System.out.println("systemTime=" + response.getSystemTime());
System.out.println("paymentTransactionId=" + response.getPaymentTransactionId());
```

#### Response Example :

```
conversationId=123456789
locale=tr
```

```
status=success  
errorCode=null  
errorMessage=null  
errorGroup=null  
systemTime=1445241016138  
paymentTransactionId=1546
```

#### JSON format of request example :

```
{  
  "locale" : "tr",  
  "conversationId" : "123456789",  
  "paymentTransactionId" : "1546"  
}
```

#### JSON format of response example :

```
{  
  "status" : "success",  
  "locale" : "tr",  
  "systemTime" : 1445241016138,  
  "conversationId" : "123456789",  
  "paymentTransactionId" : "1546"  
}
```